MECHANICAL ERROR ANALYSIS OF TELEOPERATED ROBOTS

A Thesis Submitted

in Partial Fulfilment of the Requirements

for the Degree of

MASTER OF TECHNOLOGY

by

RAVINDRA P. TELANG

to the

DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

FEBRUARY 1993

CERTIFICATE

This is to certify that the present work MECHANICAL ERROR ANALYSIS OF TELEOPERATED ROBOTS by shri Ravindra P. Telang has been carried out under our supervision and that it has not been submitted elsewhere for a degree.

Dhande

Prof. S. G. Dhande
Head, Mechanical Engineering
I. I. T. Kanpur

Sudipto Mukherjee

Dr. S. Mukherjee
Ass. Prof. Mechanical Engg.
I. I. T. Kanpur

ME-1993-M-TEL-MEC

ACKNOWLEDGEMENT

I would like to express my gratitude to prof. S. G. Dhande and Dr. S. Mukherjee for their guidance and support during the course of this work. They gave me complete freedom and constant encouragement to bring out my ideas effectively. Parental care and affection by Dr. Dhande and a friendly attitude by Dr. Mukherjee made this arduous task very enjoyable.

I wish to thank Mr. R. M. Jha and Mr. Prem Prakash of Manufacturing Sc. Lab who helped me in realizing the Master Arm in time.

Thanks are due to Mr. S. Sen and all others in Robotics Center who were always ready to help me in difficult situations.

I must thank Atul, Anirban and all of my friends who helped me in preparing my thesis and made my stay at I. I. T. immemorable.

Lastly, this work would not have been possible without the support of my family and friends. They gave me encouragement and inspiration throughout my stay at I. I. T. Kanpur.

CONTENTS

Certificate	ii
Acknowledgement	iii
Nomenclature	vi
List of figures	vii
List of tables	ix
Synopsis	x
CHAPTER 1. INTRODUCTION	1
1.1 Introduction	1
1.2 Robot calibration	2
1.3 Teleoperated robots	4
1.3.1 Mechanical Master-Slave manipulator	5
1.3.2 Powered unilateral control	5
1.3.3 Bilateral servo control	8
1.3.4 MA2000 slave manipulator	9
1.3.5 Robot controller	12
1.4 Literature review	12
1.5 Objectives, scope and limitations of present work	14
1.5.1 Objective and scope	
1.5.2 Limitations	
CHAPTER 2. DESIGN AND FABRICATION OF THE MASTER ARM	18
2.1 Parameters for Mechanical design of a robot	18
2.2 Critical guidelines for design of the	

master arm 21
2.3 Design of links and joints 23
2.4 Joint limits 30
2.5 Sensors 31
CHAPTER 3 ALGORITHM FOR ERROR ESTIMATION 32
3.1 The kinematic model 32
3.1.1 The rotation and homogeneous matrices 33
3.1.2 The link and joint parameters 34
3.1.3 The Denavit-Hartenberg representation 37
3.2 The jacobian matrix 38
3.3 Taylor series expansion 40
3.4 Error model 40
3.4.1 The minimum norm solution 42
3.4.2 Iterative updating procedure 43
3.5 Discussion 44
CHAPTER 4 IMPLEMENTATION OF THE ALGORITHM AND NUMERICAL VERIFICATION 45
4.1 Program for parameter estimation 45
4.2 Functions 49
4.3 Numerical verification through examples 50
4.3.1 Example 1 53
4.3.1.1 Observations 56
4.3.2 Example 2 56
4.3.2.1 Observations 56
4.3.3 Example 3 60
4.3.3.1 Observations 61
4.4 Discussion 62

CHAPTER 5 CALIBRATION OF MA2000 MASTER-SLAVE SYSTEM 64

5.1 Calibration of a master-slave teleoperated system 65

5.1.1 A typical position control strategy 65

5.1.2 Kinematic calibration procedure 66

5.2 Calibration of MA2000 master-slave system 68

5.2.1 Control MA2000 master-slave system 68

5.2.2 Error identification 70

5.3 Discussion 73

5.4 Conclusion 77

5.5 Suggestions for further work 77

REFERENCES 79

APPENDIX I : SPECIFICATIONS OF MA2000 ROBOT 81

APPENDIX II : EVALUATION OF JACOBIANS USING SCREW COORDINATE NOTATION 84

APPENDIX III : PSEUDO-INVERSE 86

NOMENCLETURE

$\hat{p}$	Position vector
θ	Joint angle (D-H parameter)
α	Skew angle between joint axes
d	Link offset
a	Link length
$\hat{x}$	End-effector position vector
T^E	Tool transformation
$J(\phi)$	Jacobian matrix w.r.t. variable ϕ
J_θ	Jacobian w.r.t. θ
J_α	Jacobian w.r.t. α
J_a	Jacobian w.r.t. a
J_d	Jacobian w.r.t. d
$H(\phi)$	Hessian matrix w.r.t. variable ϕ

LIST OF FIGURES

		Page No.
Fig 1.1	Cable transmission for 5 dof mechanical master slave manipulator.	6
Fig 1.2	Servo for Unilateral control.	7
Fig 1.3	Position servo for Bilateral control	7
Fig 1.4	MA2000 Slave robot.	10
Fig 1.5	MA2000 Joint motion limits.	10
Fig 1.6	System configuration MA2000.	11
Fig 2.1	The master arm.	22
Fig 2.2	Assembly drawing of the master arm.	24
Fig 2.3	Waist joint assembly	25
Fig 2.4	Shoulder joint assembly	27
Fig 2.5	Elbow joint assembly	29
Fig 2.6	Wrist assembly	30
Fig 3.1	Link and joint parameters	35
Fig 3.2	Joint axes, joint variables and link parameters (a) joint i is revolute (b) joint i is prismatic	36
Fig 4.1	Flow chart of the program for parameter estimation.	47
Fig 4.2	Kinematic diagram of robot as example 1.	51
Fig 4.3	Kinematic diagram of robot as example 2.	54
Fig 4.4	PUMA 560 : (a) schematic diagram	57
	(b) Kinematic diagram.	57
Fig 5.1	Master Waist Potentiometer Characteristics	71
Fig 5.2	Master Shoulder Potentiometer Characteristics	71
Fig 5.3	Master Elbow Potentiometer Characteristics	71
Fig 5.4	Slave Waist Potentiometer Characteristics	71
Fig 5.5	Slave Shoulder Potentiometer Characteristics	72

Fig 5.6 Slave Elbow Potentiometer Characteristics 72
Fig 5.7 Error Variation of Waist Joint 74
Fig 5.8 Error Variation of Shoulder Joint 75
Fig 5.9 Error Variation of Elbow Joint 76

LIST OF TABLES

TABLES		Page Number
Table 4.1	Denavit-Hartenberg Parameters for robot in example 1	51
Table 4.2	Parameter error chosen for robot in example 2	51
Table 4.3	End-effector position with erroneous parameters	52
Table 4.4	Convergence of algorithm for two d.o.f. robot	52
Table 4.5	Parameter errors identified for first case	52
Table 4.6	The D-H parameters for robot in example 2	54
Table 4.7	Errors on the nominal parameters	54
Table 4.8	End-effector position with errors in D-H parameters	55
Table 4.9	Convergence of algorithm for second case	55
Table 4.10	Estimated errors on nominal parameters	55
Table 4.11	The D-H parameters for PUMA	58
Table 4.12	Errors chosen on nominal D-H parameters	58
Table 4.13	End-effector position with erroneous parameters for different joint variables	58
Table 4.14	Convergence of algorithm for PUMA	59
Table 4.15	Estimated parameter errors	59

SYNOPSIS

Error analysis and calibration are essential to increase the robot positioning accuracy. The improvement in accuracy can be achieved if an accurate model of the kinematic structure of the robot is available. The actual values of the kinematic parameters differ from the nominal values due to mechanical errors. The improved kinematic model can be obtained if the errors on nominal parameters is estimated. The data available from the physical system is the measured end-effector position. An algorithm is presented to estimate the parameter errors if the measured end-effector position is known. The algorithm is implemented and numerical verification results are presented for different robots.

Teleoperated robots extend a person's manipulative, perceptual and cognitive skills to a remote location. Such a system has two working modules. Mechanical errors creep in both the modules namely the master and the slave arm. These errors accumulate and cause a significant deviation in the end-effector position of the slave arm. If a visual feedback is available then small errors can be compensated by skilled operators using his manouvaring ability. But for many applications the task to be done may not provide such a flexibility. An attempt to identify the errors in the system at joint level is made. A master arm was designed and fabricated as a part of the present work. A study of guidelines for mechanical design of robots is also presented.

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Robot positioning is measured in terms of accuracy and repeatability. Robot position accuracy refers to the ability of a robot to move to a commanded location. This commanded location would not have been taught to the robot but would be specified in terms of workspace world coordinates. Repeatability refers to the robot's ability to return to a given position once it has been taught.

For the implementation of industrial robots in an integrated manufacturing environment, it is necessary to be able to position the end-effector with high accuracy. Position errors of the order of 10 mm are not unusual in present industrial robots [9]. They, however, have good repeatability of the order of 0.1 mm. Robot calibration seeks to improve on the robot positioning accuracy.

For high accuracy, an accurate model of the kinematic structure of the robot is essential. Accurate control of the joint variables yields high repeatability. Calibration is the process of estimating the exact kinematic model of the robot. The parameters usually have small deviations from the design or nominal values. Though these errors may be small individually, they cause a

significant change in the end-effector position due to serial chain nature of most industrial robots. Calibration minimizes the risk of having to change an application program due to slight changes or drifts, such as wear of parts, dimensional drifts, tolerances and component replacement effects in the robot system. Calibration process can be divided into four steps as Modeling, Measurement, Identification and Correction.

Most of the literature regarding calibration highlights the error synthesis of industrial robots. Calibration techniques for special purpose robotic systems such as teleoperated robots are also needed. If such a system is provided with sensory feed back such as visual feedback, the end-effector positioning errors can be compensated by skilled operators using manouvering skills. But for many applications such a feed back may not be available. Even if it is available, the task to be done may not provide such flexibility. In all such cases, calibration becomes a must.

1.2 ROBOT CALIBRATION

Robot calibration involves identifying a more accurate relationship between the joint transducer readings and actual workspace position of the end-effector. The identified changes are used to permanently change the robot positioning software. Major cause of inaccuracy in the robotic system is due to the difference

between nominal geometry of robot, based on robot design specifications, and the real geometry. This difference may be attributed to the manufacturing tolerances, mounting errors during robot link assembling and kinematic model simplifications in the control unit. Nominal models used are simple and are based on several assumptions, such as parallelism or orthogonality of the axes of the joints. Difference between nominal and real geometry also arise from nongeometrical errors such as link and joint flexibility, backlash, gear run out and the like.

Robot kinematic calibration consists of identifying a more accurate geometrical relationship between joint encoder readings and the actual position of an end-effector, based on the nominal relationship between them.

Robot calibration can be divided into 3 levels [11]. Level 1 calibration can be called as joint level calibration. The goal here is to determine the correct relationship between the signal produced by the joint displacement transducer and the actual joint displacement. Level 2 calibration may be defined as entire robot kinematic model calibration. The objective here is to determine the basic kinematic geometry of the robot as well as correct joint angle relationship. Level 3 calibration can be defined as nongeometric calibration. Non-kinematic errors in positioning of the end-effector of a robot are due to effects,

such as joint compliance friction and clearance, as well as link compliance. This type of calibration also involves changes in the dynamic model if robot is under dynamic control.

Any calibration process consists of four steps. The first step is to choose the form of a suitable functional relationship. It is referred to as modeling step. Second step is to collect some data from the actual robot that relates the input of the model to output. This is termed as measurement step. The third step is the data processing which leads to identification of certain coefficients in the model. The fourth and the final step is to modify the nominal model and incorporate the changes in position control software. This step is termed as the correction step.

1.3 TELEOPERATED ROBOTS

Teleoperation is the extension of a person's manipulative, perceptual and cognitive skills to a remote location. The increased need for remote manipulation in environments hazardous to human operators such as nuclear, under sea, outer space etc., has been widely recognized. Many tasks in such environments are complex, unpredictable and unplanned and therefore cannot be done by the present generation of machines. A teleoperator couples human decision making and motor control

functions with a manipulator working in a remote or hostile environment. The aim of teleoperation is to give the user charge of the movements of remote manipulator. This can be done in three ways -

1. Mechanical master-slave telemanipulators
2. Powered telemanipulators with unilateral control
3. Powered telemanipulators with bilateral control.

1.3.1 Mechanical Master Slave Manipulators

In these types of manipulators the motion of the operators hand are reproduced by mechanical linkages connecting a slave arm with a gripper to a master arm ending in hand control. The mechanical connection allows the user to feel the forces on the slave and so there is some feedback. Fig. 1.1 shows the principle of cable transmission for 5 degrees of freedom of such manipulator, the sixth being produced by rotating the whole system around a sleeve through the wall.

1.3.2 Powered-unilateral control

In unilateral control, there is no force feedback. The operators control device may take the form of a master arm whose joint angles are measured and which drives the corresponding joint servos in the slave arm. They are used wherever mobility is required. A servo for unilateral control of one joint is shown in Fig. 1.2. It is an open loop in that there is no feedback from the

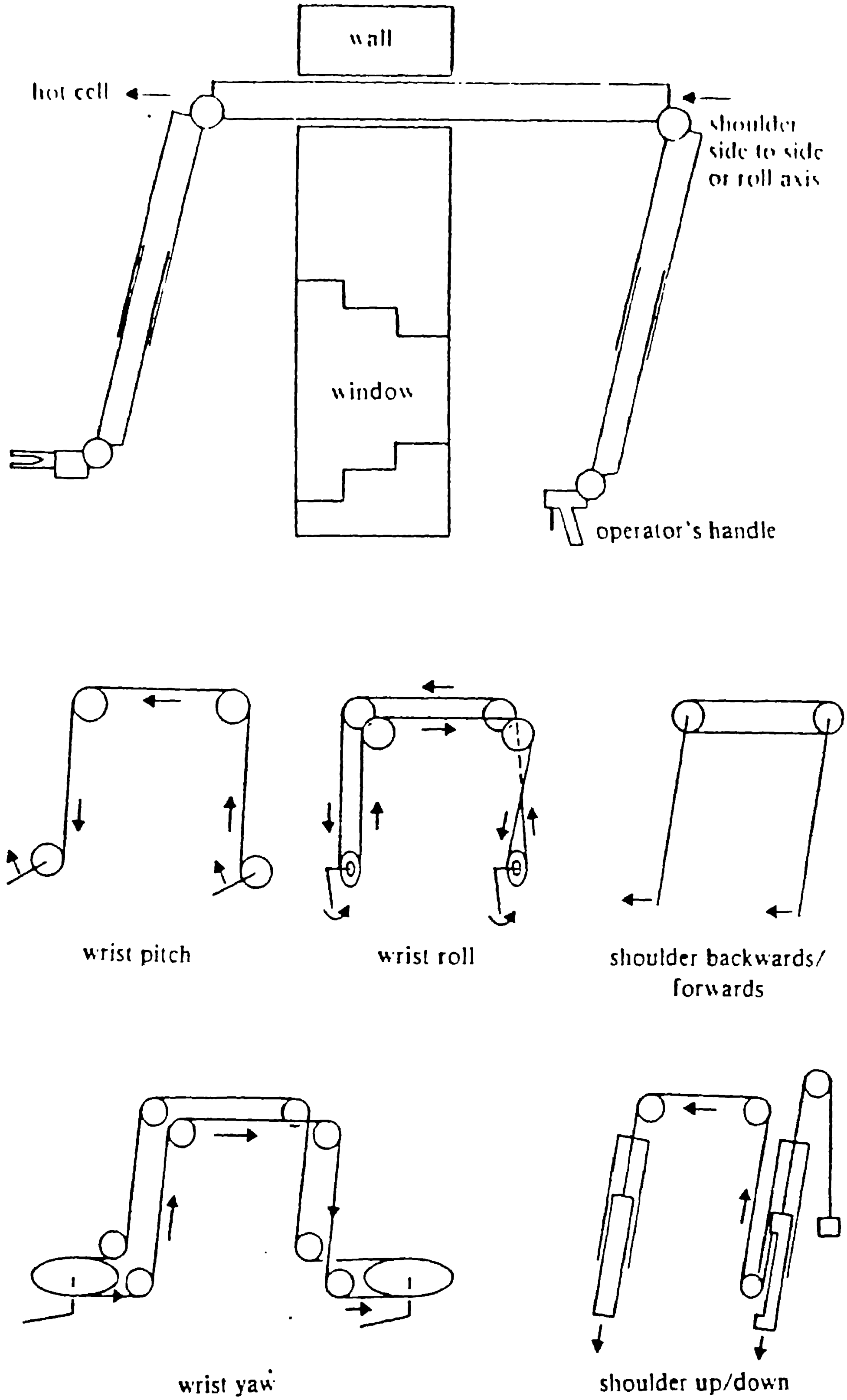


Fig 1.1 Cable transmission for 5 dof mechanical master slave manipulator.

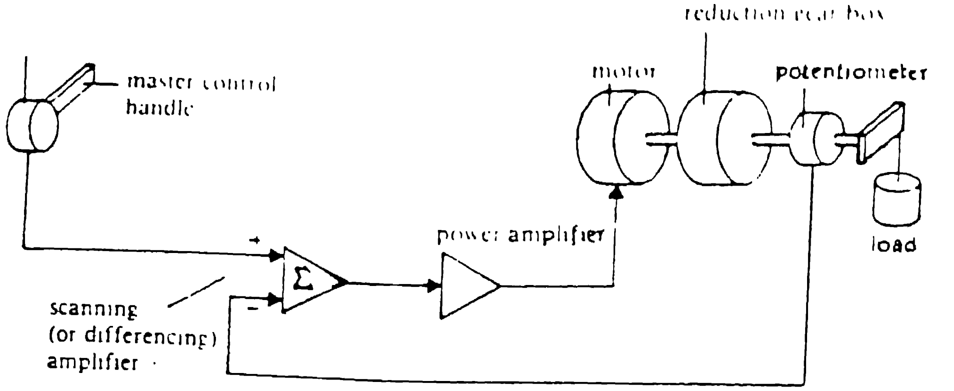


Fig 1.2 Servo for Unilateral control.

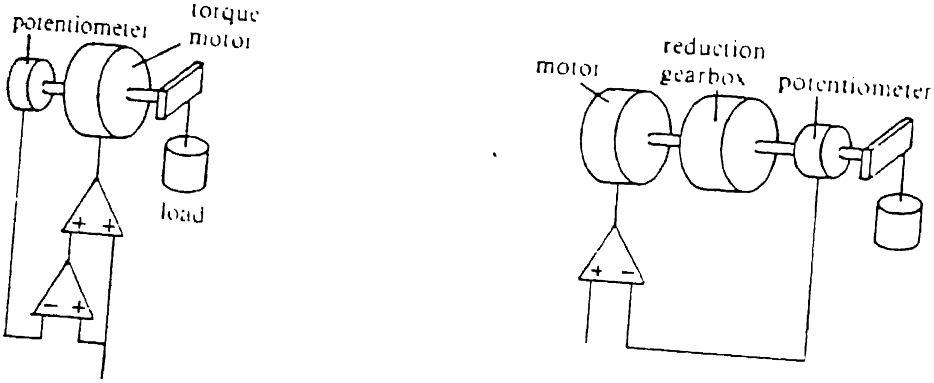


Fig 1.3 Position servo for Bilateral control

slave to the master, but there is a position feed back within the slave half of the system.

1.3.3. Bilateral Servo Control

In this type of control the forces on the slave arm is fed back to the master arm for the operator to feel. Force reflection helps the operator to consider himself present in the remote location and thus induces *telepresence.* Systems with telepresence are symmetrical where master can also be moved like slave just as in case of mechanical master slave manipulators. There are many types of bilateral control systems. The master can be designed to take position or force as its input, the controlled variable at the slave also being position or force and similarly for the return loop. Actuators are provided at the master end as well to enable it to be back driven and torque motors generally act as transducers of torque. At the master and slave ends, position or force servos can be used. A position servo (Fig.1.3) maintains the output shaft angle equal to the input commanded angle regardless of load. A force servo maintains the output torque proportional to the commanded torque regardless of position. Some torque opposes the torque produced by the motor so as to stop the shaft from moving. When this happens the signal from torque transducers balance the input signal and the shaft does not turn. The summing amplifier is required, since the motor

at master end is back-drivable also and so even if there is no error minimum torque must be kept. In other cases gears can support the load and motor current can be zero.

1.3.4. MA2000 Slave Manipulator [13]

MA2000 consists of a robot arm, electronic controller, teach key pad and associated software. Robot arm has six d.c. motor powered revolute joints operating with closed loop control. It also has a pneumatic gripper. The specifications of the robot are shown in Appendix I. A schematic diagram of MA2000 is shown in Fig. 1.4. MA2000 operates with IBM compatible host computer. The key pad is used to communicate with the computer and controller. It has additional input/output ports for 100 steps of joint position data. It has following programming options :

1. Lead by nose - continuous path.
2. Lead by nose - point to point.
3. Teach key pad - point to point.
4. Off line computer programming.

In all the above options it works on teach and playback mode. Each step has information regarding step number, rate of movement (0-9 discrete steps), input, output, wait (time in seconds), jump (to a certain step), the joint parameters for waist, shoulder, elbow, pitch, yaw and roll, and gripper status (close/open). All joints are allotted micro motion steps in the

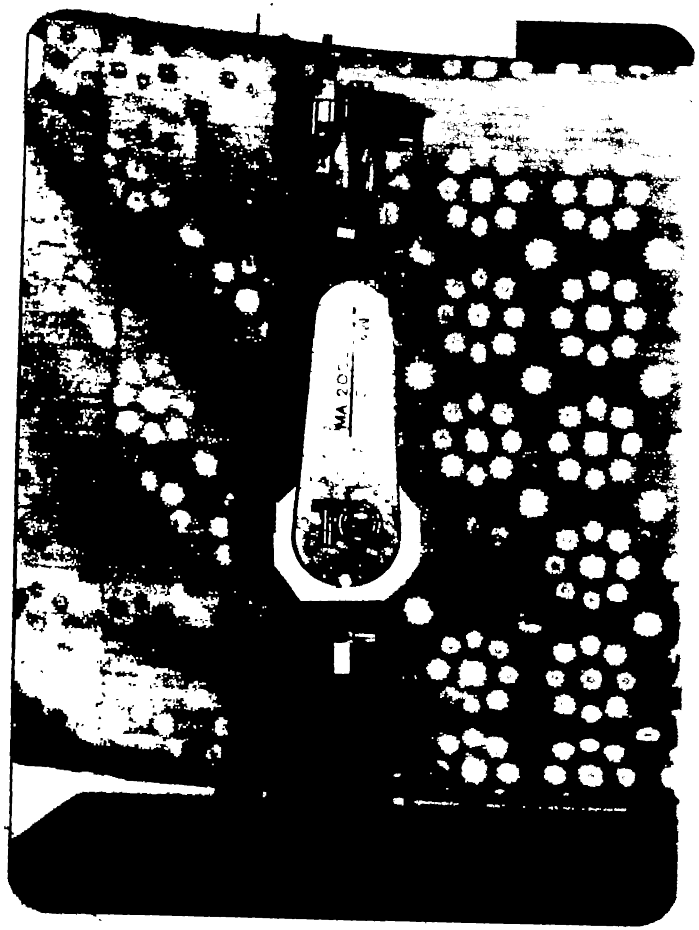
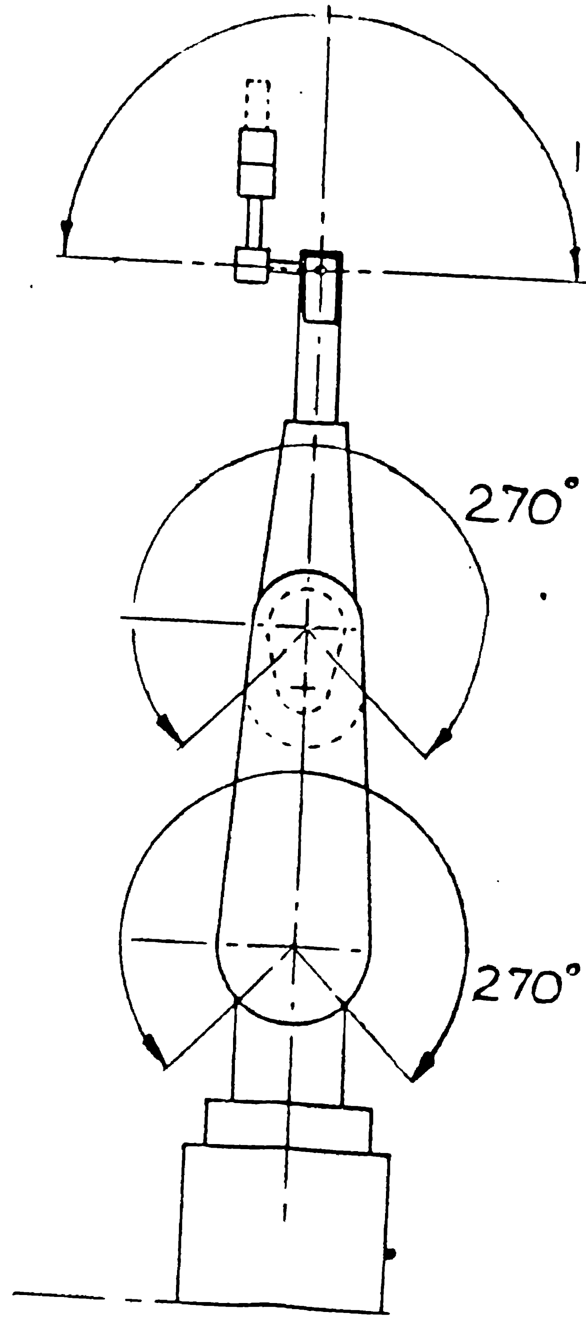


Fig 1.4 MA2000 Slave robot.



Other Joints

Waist : 270°

Yaw : 180°

Roll : 180°

Fig 1.5 MA2000 Joint motion limits.

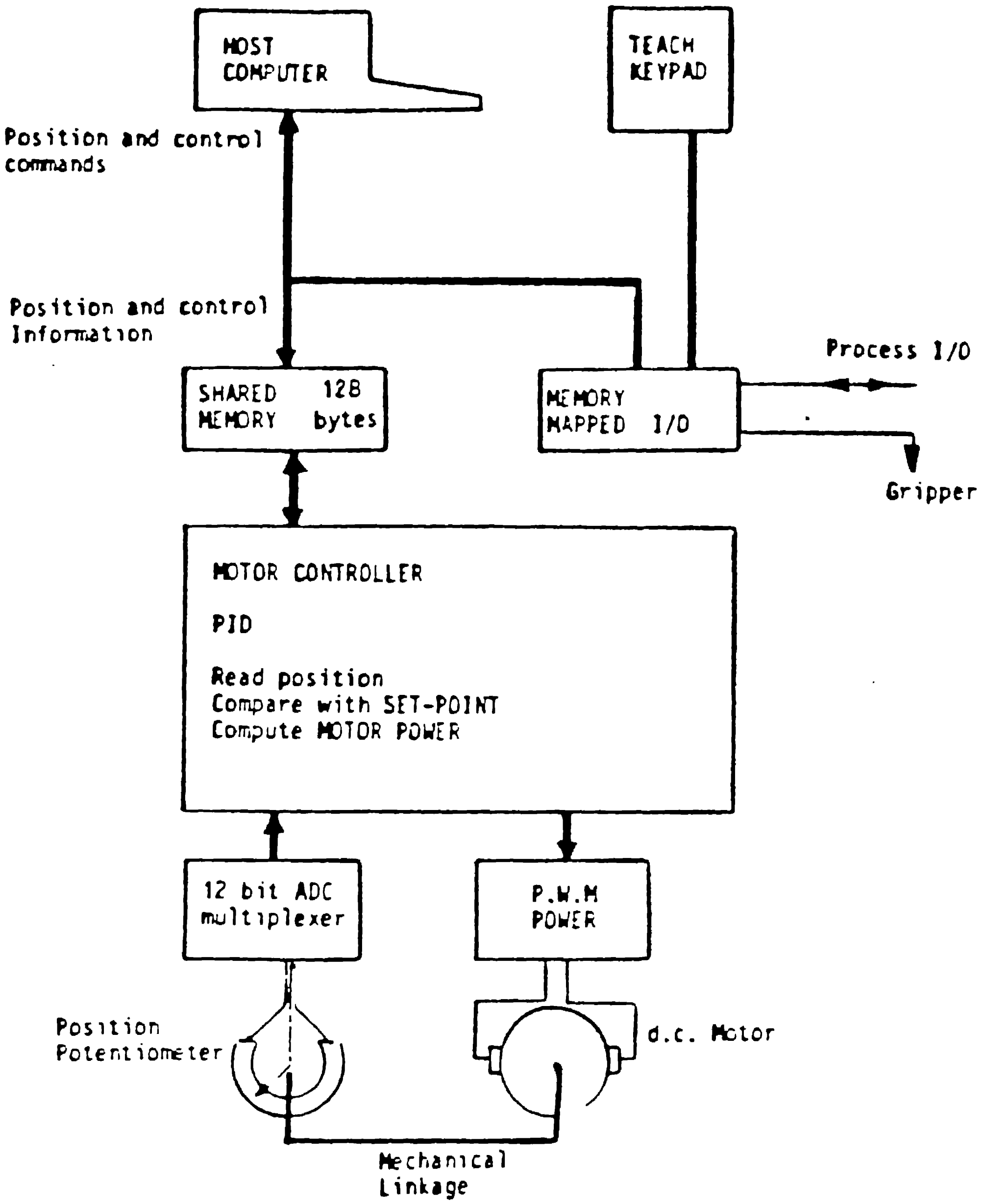


Fig 1.6 System configuration MA2000.

range 0 - 999, within it's limits of motion. The limitation of joint motion is shown in Fig.1.5. The robot can be taught only in lead by nose-continuous path mode.

1.3.5. Robot Controller

The controller of MA2000 consists of a combined interface and motor controller board. It is designed to be driven from the host computer which supplies position information for each of the robot joints. The controller will then supply PID digital servo control of up to 6 channels to control the robot axes to achieve the required joint positions.

Motor controller uses a 6502 CPU operating at 1M Hz clock rate. It's prime task is to perform a real time PID control. It also communicates with the host computer to obtain position information and interrogate the front panel controls. The system block diagram is shown in Fig. 1.6.

1.4 LITERATURE REVIEW

Robot position error models have been widely studied. Most error models in the literature attempt to relate the perturbations in the robot kinematic parameters to the differential change in the robot tool pose. A linearized geometric model has been established by Veitschegger and Wu [8]. This linearized model was also experimentally verified by implementing

it on PUMA 560 by them [9]. This model introduces a rotation around Y-axis. [Rot(y, β)] into homogeneous transformation that would allow small errors in consecutive parallel axis. Menq and Borm [2] have used similar formulation for their study of statistical properties of the position error of the end-effector. They have employed probabilistic characterization of the unknown geometrical parameters. But they have not dealt with identification, measurement and calibration problem. Roth et al. [10] have given an overview of robot calibration discussing all the four aspects modeling, measurement, identification, and correction. They however have not presented any experimental results. Hollerbach and Bennet [4] have used Veitschegger's linearized error model for geometrical parameter identification. They have exploited the redundancy with respect to task degrees of freedom to eliminate the need for end-effector position measurement. Renders et al [5] have built a maximum likelihood estimator around the same linearized model for identification of geometrical errors. Experimental verification results were also presented for a six degrees of freedom robot. A very important aspect of robot calibration is the measurement of end-effector position. A large number of different experimental setups are presented in the literature. Mooring and Padavala [1] have used a Coordinate Measuring Machine with touch probe and a specially

designed end effector to calibrate a PUMA robot by measuring the complete tool pose at each observation. Judd and Knasinski [7] have used two theodolite triangulation to calibrate an Automatic A10900 robot. The method used a specially designed end-effector. Different types of calibrated fixtures can also be used to measure position information during robot calibration. Veitschegger and Wu [10] have used a fixture plate with a set of precisely positioned holes and an end effector with a positioning device. Morris and Pathre [6] have used a vision-based automatic theodolite (VBAT) for partial pose measurement. With Automatic tracking, focusing and centering a high degree of repeatability is claimed. Experimental results presented in the work show that the model could predict the target position to a root of mean square miss-distance of 0.2 mm.

Not enough literature is available so far regarding application of calibration techniques to the teleoperated robotics systems. However, telerobotics in general has generated a lot of interest in the researchers. [12] and [16] have given an in depth description of the issues in tele-robotics. Concepts of master slave teleoperation, unilateral and bilateral controls have been explained and state-of-the-art technologies are presented.

1.5 OBJECTIVES, SCOPE AND LIMITATIONS OF PRESENT WORK

1.5.1. Objectives and Scope

Industrial robots in general show poor accuracy. A large amount of position error is introduced at the end-effector due to the manufacturing tolerances which result in a change in the nominal geometry of robots . These changes in the nominal geometry are difficult to measure accurately. But the end-effector position can be measured with a reasonable accuracy. This accurate enough value of end effector position can be used to determine the correct values of different parameters. To obtain a closed form solution is however practically impossible. A typical inverse kinematic solution for joint angles of a six degrees of freedom robot manipulator of nominal geometry requires 12 transcendental function calls, 34 multiplications and 14 additions. Hence to estimate the geometrical parameters use of numerical techniques is unavoidable.

Our primary objective in the present work has been to develop an algorithm to estimate the parameter errors on the Denavit - Hartenberg link and joint parameters. This estimate of parameters is then used to calibrate the robot. This method of calibration eliminates the need of creation of an error map of the workspace of a robot. A few accurately measured points can be used to estimate the parameter errors. The assumption being that

errors on nominal parameters are not dependent on the posture of the manipulator. This assumption in general is true as precisely manufacture and accurately assembled manipulators are likely to be free from all sorts of flexibilities and joint plays. The improved geometric model can then be used to accurately solve the forward kinematics for the end-effector position. Using that corrected joint angle values can be determined by Jacobian control techniques to significantly improve the inverse kinematic solution.

This method is implemented to calibrate MA2000 master slave teleoperated manipulator. A master robot for the teleoperated system was designed and fabricated as a part of present work.

An algorithm for parameter estimation has been developed keeping in mind the six degrees of freedom robot, but it is flexible enough to be used for robots with different number of degrees of freedom. Numerical verification of the algorithm is done for three different types of robots.

1.5.2. Limitations

Denavit - Hartenberg kinematic model is the most popular approach for forward and inverse kinematics. But for the purpose of error analysis and calibration it has a serious limitation. This problem occurs when neighboring joint axes are parallel. The

common normal between them is extremely sensitive to slight variations in axes orientation. Another problem with the procedure is that sometimes singularities in the parameters may be encountered that may hinder the iteration. This is particularly likely to occur in case the initial guess of the nominal parameters is far from the correct values.

Besides these inherent shortcomings of the procedure the accuracy obtained is limited by the accuracy of measuring device. We have used travelling microscope for this purpose. Despite these limitations a reasonably good accuracy is expected on the end effector position. According to Randers et al [5]an error of 0.5 mm is the best result that can be expected without considering nongeometrical errors.

CHAPTER 2

DESIGN AND FABRICATION OF THE MASTER ARM

For employing the MA2000 robot in teleoperator mode, it was essential to develop a man- machine interface with primary controls achieved through the use of hand controls. The master arm which would accomplish the task of interfacing must have six degrees of freedom, three for exact positioning and rest for orientation purpose. These degrees of freedom should facilitate positioning of end-effector in any part of workspace in a desired attitude. The master arm should be kinematically equivalent to the slave arm.

This chapter discusses guidelines followed for the design of master arm and some of the basic parameters which must be considered while designing a robot. These parameters for the master arm were guided by the slave arm hence their selection procedure has not beeen explicitly stated here.

2.1 PARAMETERS FOR MECHANICAL DESIGN OF A ROBOT

Design of a robot is a challenging task as basic six degrees of freedom can be achieved through many variations. This makes optimum design an illusive target. Variability of robot

functions and the need for programmability makes it difficult to define a unique set of specifications.

Despite the diversity of robot applications and differences in their configuration, a basic set of parameters can be defined which are closely associated with the mechanical structure. Many of these parameters may also be interrelated.

(i) Payload.

The rated payload of a robot is the maximum mass which it can handle in any configuration of it's linkages. Load carrying capacity may be different in different configuration. Payload capacity of the slave is 0.5 Kg. As there is no force feedback to the master this parameter would be insignificant for design of master arm but in future force feedback may be provided by slightly modifying the wrist design only.

(ii) Mobility and Degrees of Freedom.

Mobility of a robot is determined by the total number of independent motions which can be performed by all it's links. These motions can be either translational along a certain axis or rotational around an axis. The slave robot has six degrees of freedom and hence the master was also designed to have six degrees of freedom.

(iii) Workspace.

Workspace of a manipulator is the space composed of

all points which can be reached by it's arm end point or some point on it's wrist but not including end-effector or tool tip. The reason for such a standard is that the workspace is a characteristic of the robot, but end-effector, tools, etc. can have various sizes and shapes. The workspace of the slave arm is a hemisphere of 500 mm centered at the shoulder joint. Workspace for the master arm was also identically designed.

(iv) Agility.

Agility means the effective speeds of execution of prescribed motions. Effective speed is not the maximum prescribed speed, but, the average speed at which end-effector would do a task taking into account the acceleration and decelerations. This parameter was not crucial for design of master as it would be moved by human operator and speed of execution would always be less than that of slave arm.

(v) Accuracy and Repeatability.

Accuracy and repeatability would influence the mechanical design critically. Repeatability would be achieved by accurate control of joint variables and accuracy can be increased by minimizing the mechanical tolerances and joint flexibilities. Repeatability was not a consideration in design of master arm as no actuators are mounted on it. Minimum possible tolerances were given on the machining and fabrication jobs.

(vi) Stiffness, Damping Coefficients and Natural Frequency.

All these parameters are critical to structural dynamics. Structure should have as high stiffness as possible. Low value lead to low natural frequencies and hence longer settling time which causes deterioration in the performance. Damping coefficient should be as high as possible as it shortens the transients and enhances dynamic stability. The master arm is very light and stiff so the natural frequency is expected to be high.Hence no attempts were made to dampen it.

(vii) Economic Aspects.

Cost effectiveness of robots would be dictated by investment costs and running expenses. Cost of the materials for the master arm was around Rs. 750.

2.2 CRITICAL GUIDELINES FOR DESIGN OF MASTER ARM.

Most of the parameters for design, as stated above, are dictated by the slave arm. Apart from the general design criteria a few parameters need to be emphasized in the design of master arm. These critical parameters are as follows

(i) Kinematic Equivalence

Master arm must be kinematically equivalent to the slave arm. This type of equivalence is essential for two reasons. First, the kinematic equivalence makes the correspondence between

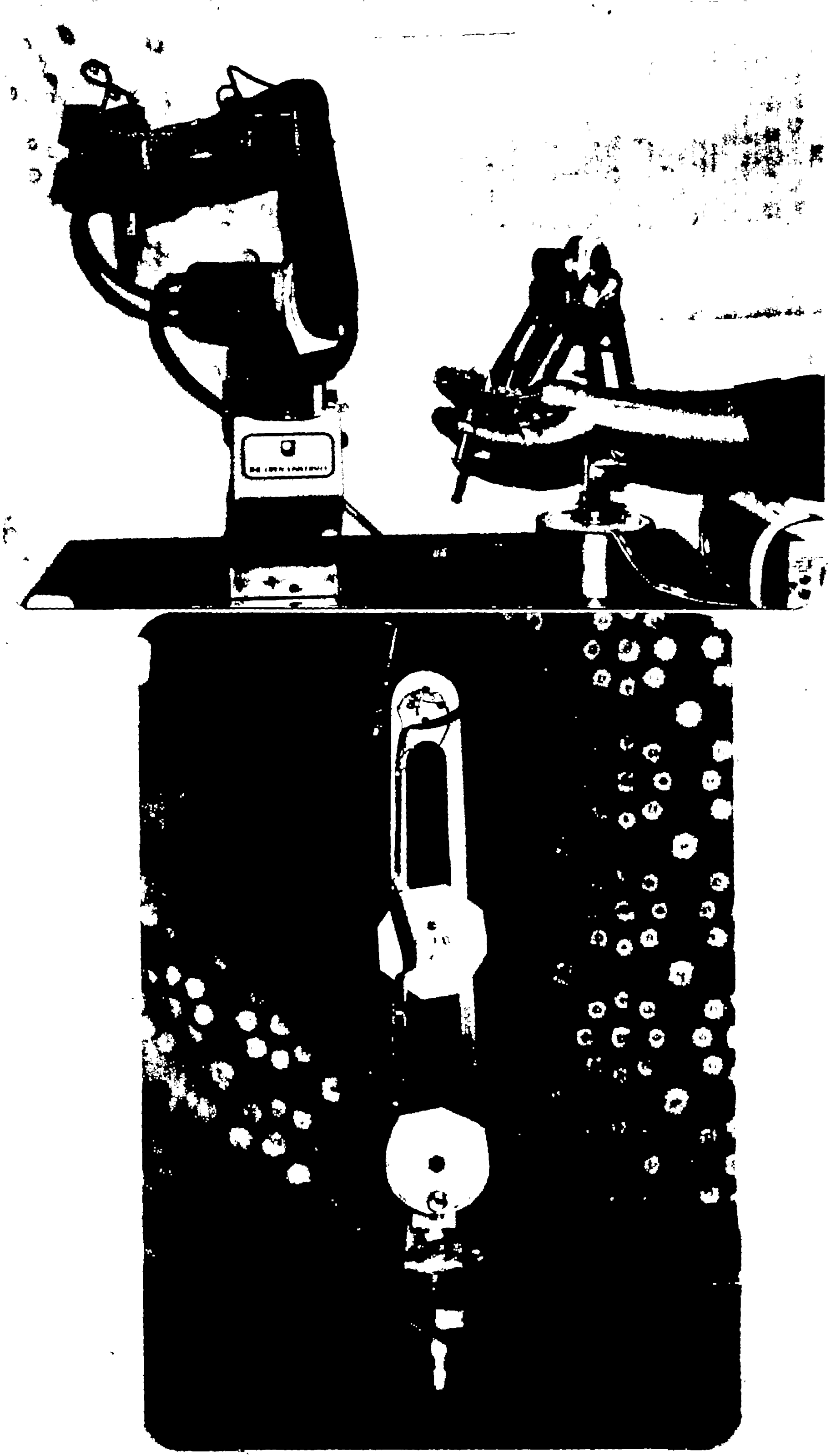


Fig 2-1

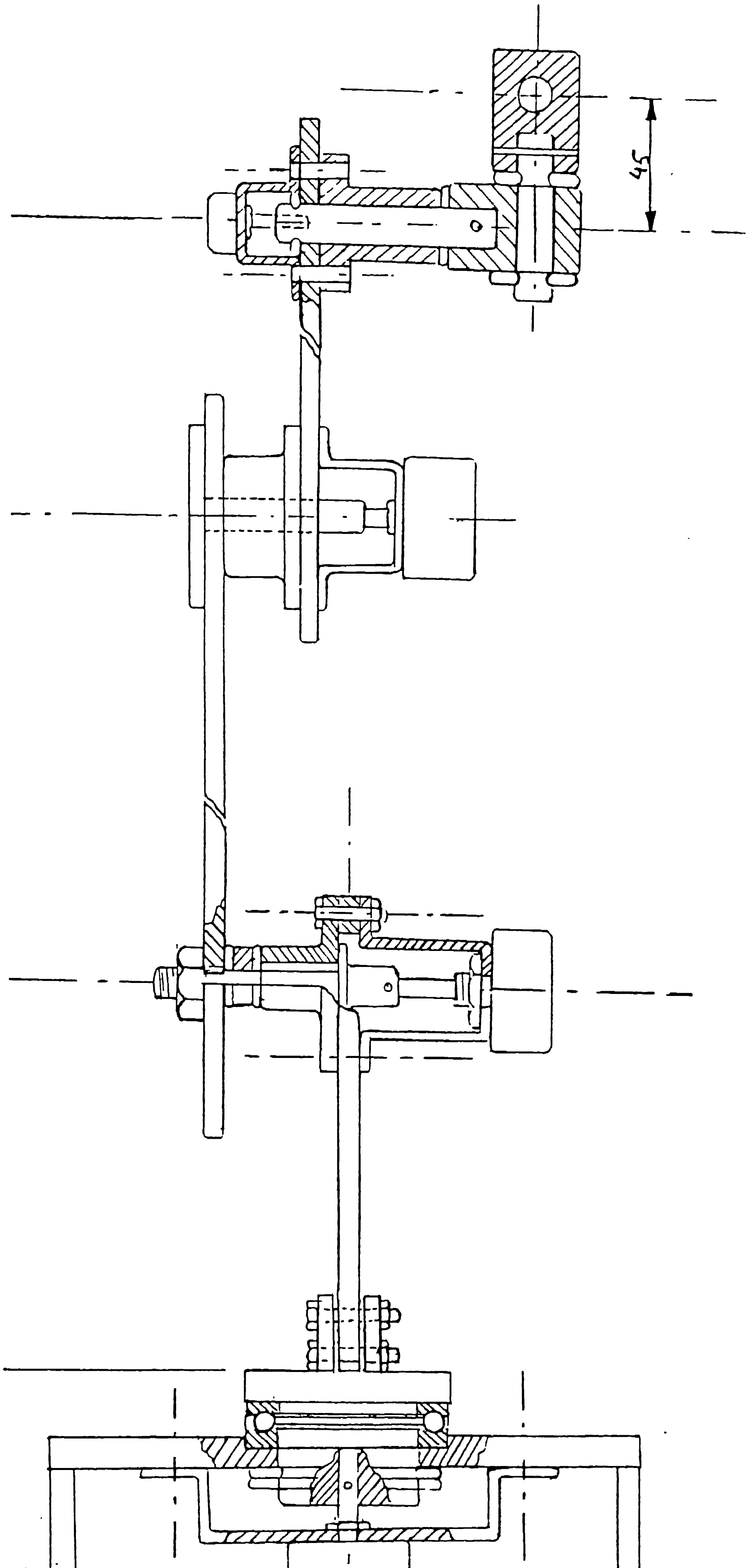
two robots self evident. If the two robots are not kinematically equivalent it will be difficult to establish the appropriate correlation between the two. The other important reason is that the control strategy for the system does not take into account the forward or the inverse kinematics of master and slave robots. Operation of the system depends on the joint transducer readings of master being transferred to the slave after appropriate scaling. Hence if end-effector of two robots are to be in approximately same pose, the two robots must have exact kinematic equivalence. This was ensured by having the Denavit-Hartenberg parameters to be the same for both the robots.

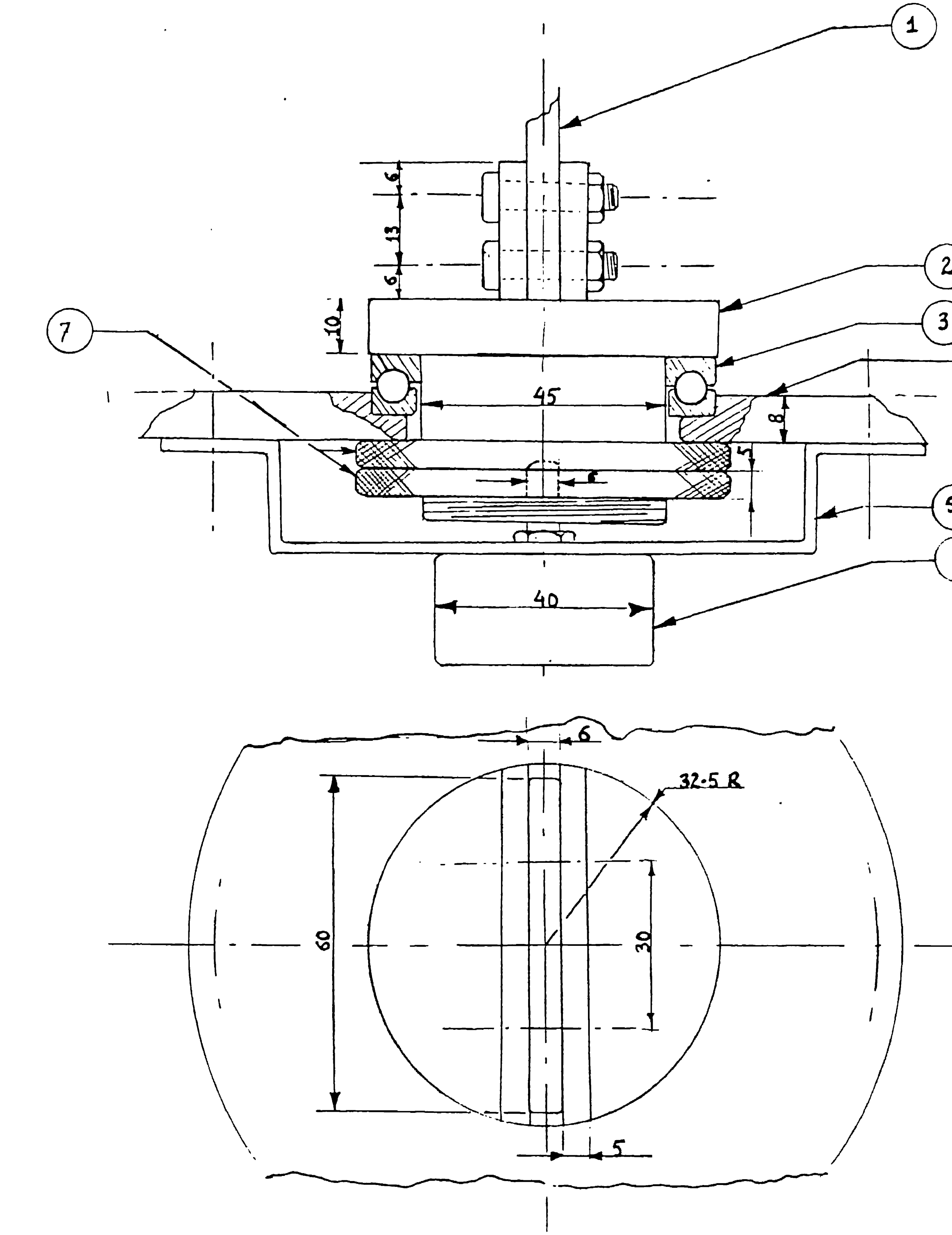
(ii) Structural Stiffness and Weight.

Since the primary purpose of fabricating the master arm was to have a better performance in terms of end-effector positioning accuracy, the master arm was designed to have minimum of joint and link flexibilities. Besides having rigidity of structure it was equally essential to have as lower weight as possible. From ergonomic criterion the effective arm weight must be less than 0.5 Kg .

2.3 DESIGN OF LINKS AND JOINTS.

The design of all links and joints was carried out in coherence with the above guidelines. Figure 2.1 shows the


45



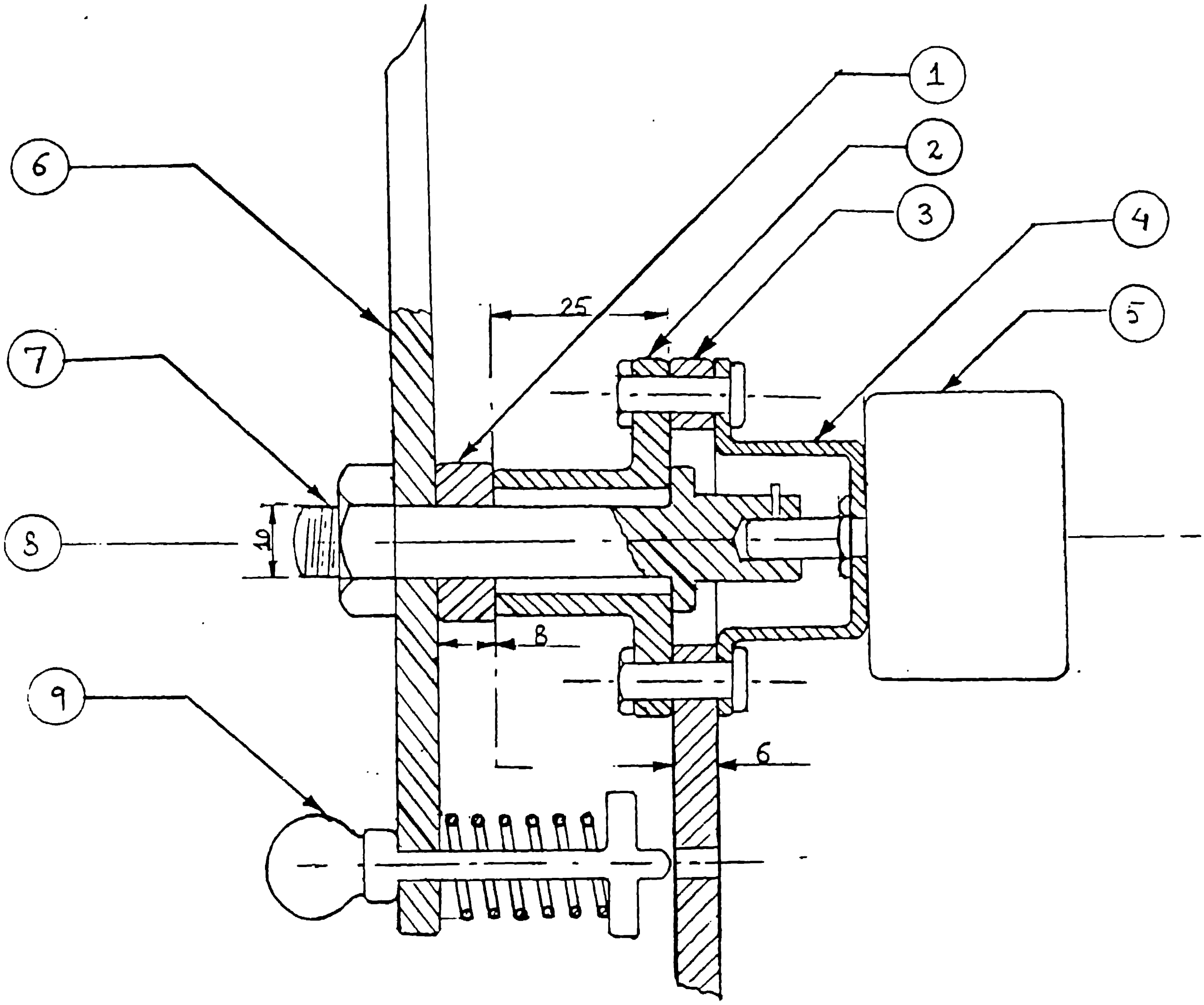
1. Shoulder link (link 1)
2. Waist joint shaft
3. Thrust bearing
4. Base link (link 0)
5. Collar
6. Potentiometer
7. Lock nuts

photograph of fabricated master arm. Figure 2.2 shows the assembly drawing of the arm.

The arm has six links. Three longer links for waist, shoulder and elbow and three smaller ones for the wrist assembly. Material selected for longer links was Aluminum and that for shorter links was Bakelite. Both of the materials are light and easily machinable. The lower three links were fabricated from 6.0 mm thick plate. To reduce weight, the excess material was removed. Remaining links were made from a bakelite rod having a diameter of 55 mm.

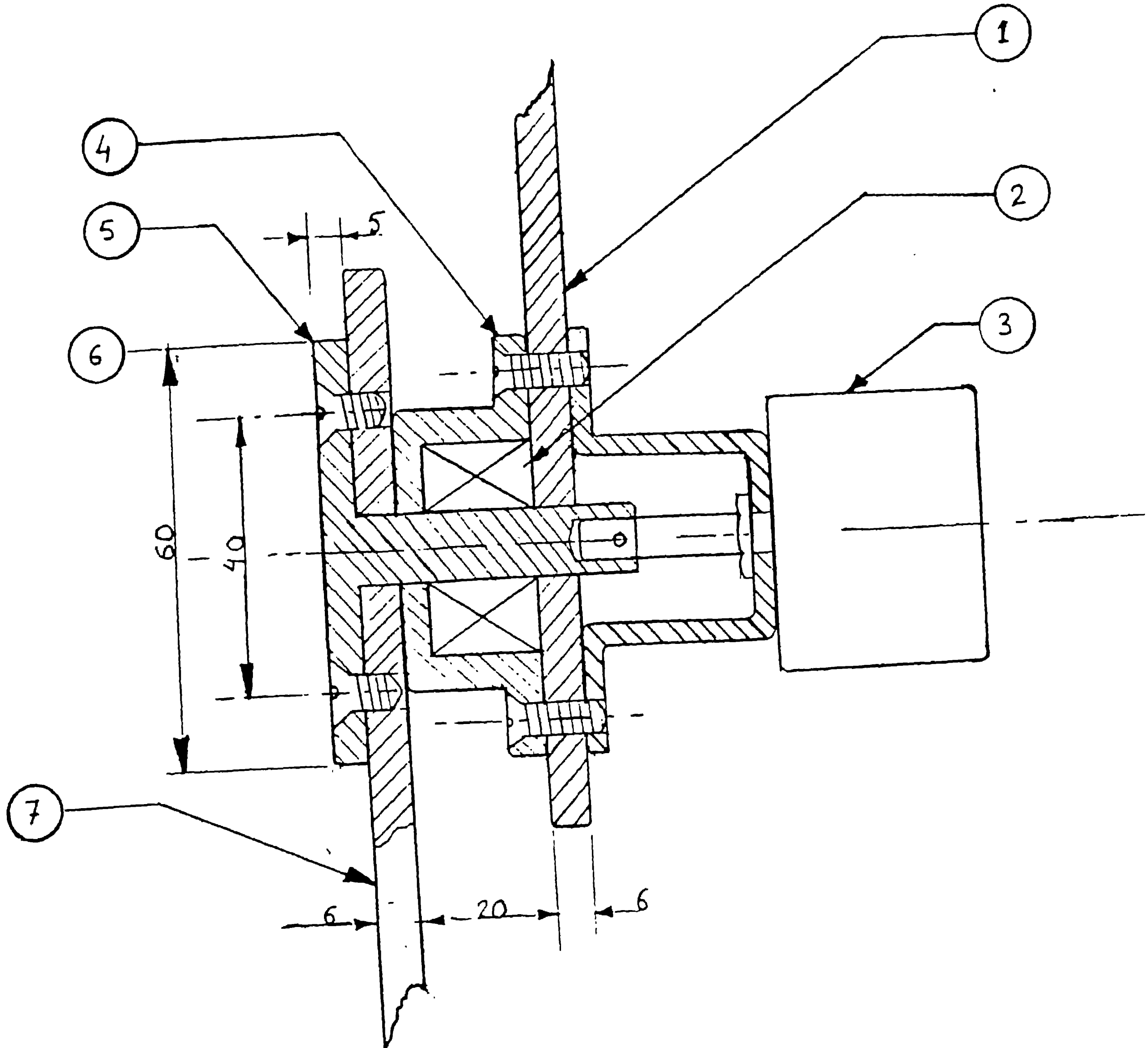
All the joints in the master robot are revolute. Appropriate bearings have been used as per requirements. Waist joint design employs a single thrust bearing with a lock nut arrangement to constrain the motion. Figure 2.3 shows the detail design of the joint. Primary consideration of the design was to avoid wobbling. It was crucial to have minimum of tolerances at this joint as errors here would get multiplied at the end-effector.

Shoulder joint assembly is shown in figure 2.4. Bush bearing has been used instead of ball bearing. This was done mainly to increase the support area, as cantilever design is incorporated here. Considerations were different for the elbow joint. At shoulder joint offset between the links and net weight



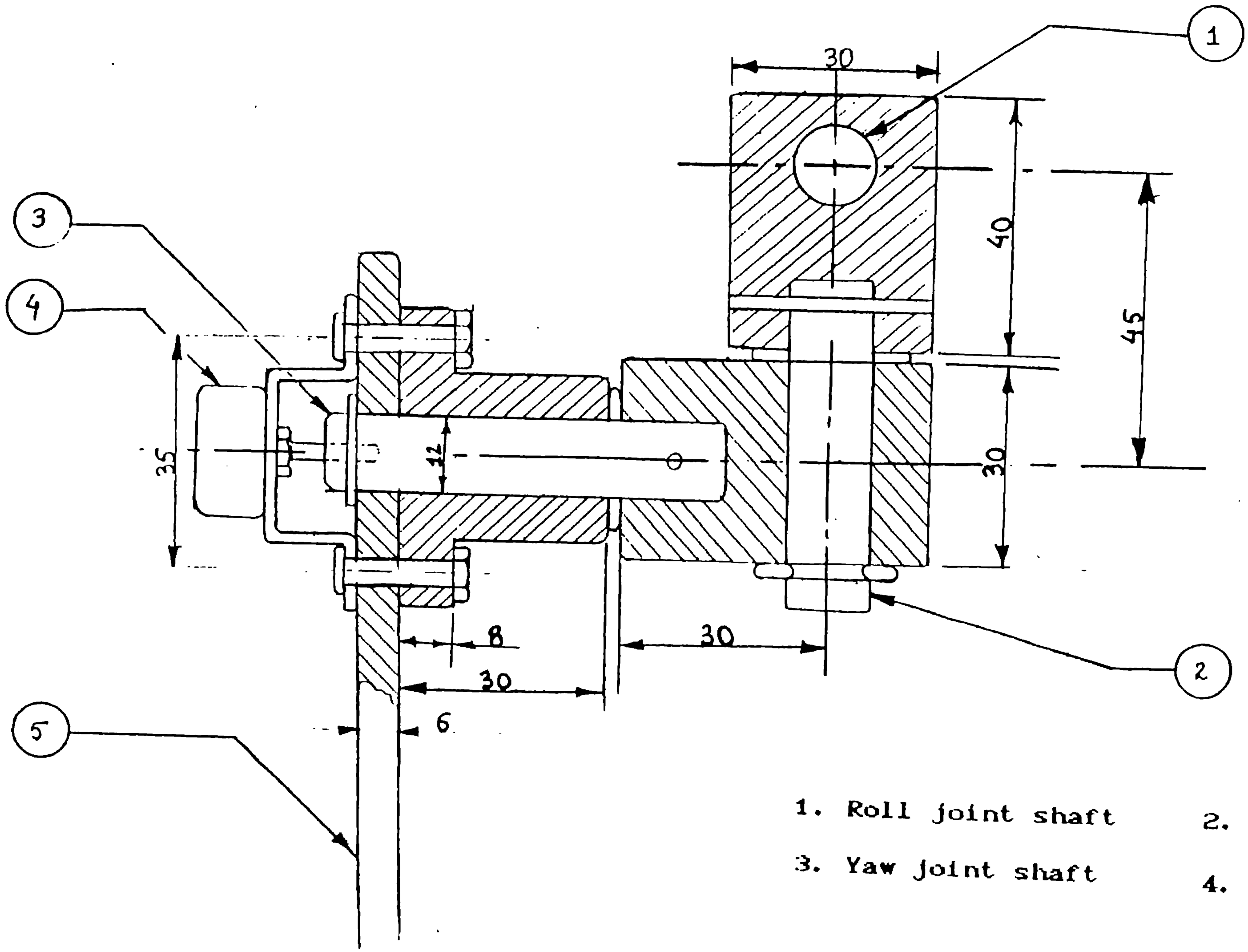
1. Bakelite spacer
2. Bearing seat
3. Shoulder link (link 1)
4. Collar for pot.
5. Potentiometer
6. Elbow (link 2)
7. Shoulder joint shaft
8. Bearing
9. Locking device

Fig 2.4 Shoulder joint assembly



1. link 3
2. Bearing
3. Potentiometers
4. Bearing s
5. Elbow shaft
6. link 2

Fig 2.5 Elbow joint assembly



1. Roll joint shaft
2. Pitch joint shaft
3. Yaw joint shaft
4. Potentiometers

Fig 2.6 Wrist assembly

acting were larger. This caused for the choice of bush bearing at the joint. The net moment at elbow joint are relatively small and ball bearing could be used Figure 2.5 shows the assembly for elbow joint.

The wrist was designed to have as small weight as possible. Bakelite was used for making links and Perspex rods of 12.0 mm diameter for making the corresponding shafts. No additional bearings were used as Perspex and Bakelite form a smooth revolute joint. Moreover the use of bearings would have increased the weight.Figure 2.6 shows the assembly for wrist joint.

2.4 JOINT LIMITS

The slave robot has limit switches at all joints and hence can execute about 270° rotation at the waist, shoulder and elbow joints and 180° rotation for roll, pitch and yaw motions. To limit the motion of the slave within above limits, mechanical stops have been provided at the corresponding positions on the master arm joints. Thus all the motions made by operator at master arm are within permissible limits and hence executable at slave end.

2.5 SENSORS

Potentiometers were used as joint position sensors as they are reliable in continuous operation and easily available. They were mounted directly on the joint axes as joint transducers. Waist, shoulder and elbow joints have three turn potentiometers and all wrist joints have single turn potentiometers. By directly mounting the potentiometers on joint axes, information loss due to slip and transmission were avoided. This in turn caused only a part of the full range of the potentiometers to be utilized.

CHAPTER 3

ALGORITHM FOR ERROR ESTIMATION

The positioning accuracy of commercially available robotic manipulators depends upon a kinematic model which describes the robot geometry in a parametric form. Manufacturing errors in the machining and assembly of manipulators lead to discrepancies between design parameters and physical structure. Unfortunately for many industrial robots, economic reality preludes the further reduction of manufacturing tolerances. It thus becomes essential to expand the kinematic model to account for these manufacturing errors. It is attempted to develop an algorithm for estimating the kinematic parameters, given the measured end-effector position.

3.1 THE KINEMATIC MODEL

A mechanical manipulator can be modeled as an open loop articulated chain with several rigid links connected in series by either revolute or prismatic joints which are driven by actuators. One end of this chain is attached to a supporting base while other end is free to which a tool (end-effector) is attached, to manipulate objects or perform assembly operations. The relative motion of the joints results in the motions of links that positions end-effector in a desired configuration. The joint position described by a set of variables called the joint variables. Configuration of end-effector in terms of position and orientation forms another set of variable. This position and

orientation of the end-effector is usually described in a fixed reference frame. The task which a forward kinematic model accomplishes is to determine the end-effector position and orientation given the joint variables.

3.1.1 THE ROTATION AND HOMOGENEOUS MATRICES

Vectors and matrices are used to develop a systematic and generalized approach to describe and represent the location of the links of a robot arm with respect to a fixed frame of reference. Coordinate frames attached to each link are established with one axis along the joint axis. Such an allocation reduces the direct kinematic problem to finding the transformation matrices that relates the body attached coordinate frames with the previous frame.

A rotation matrix [3 X 3]can be defined as the transformation matrix, operating on a position vector in a three dimensional Euclidean space, which maps its coordinates expressed in a frame attached to a a rotated body to a fixed reference frame. A rotation matrix can be constructed representing the direction cosines of principal axes of the rotated coordinate system with respect to the reference coordinate system.

This type of rotation matrix does not give us any provision for translation and scaling. To accommodate these feature in it a fourth component is introduced to a position vector $\hat{p} = (p_x \; p_y \; p_z)^T$ in a three dimensional space which makes it $\hat{P} = (\; mp_1 \; mp_2 \; mp_3 \; m \;)^T$. Thus vector $\hat{P}$ is said to be represented in HOMOGENEOUS COORDINATES. In general the representation of an N component position vector by a (N+1) component vector is called homogeneous coordinate representation. The $(N+1)^{th}$ component is

the scale factor. For robotics applications the scale factor is chosen to be one. A homogeneous transformation matrix can be considered to consist of four sub matrices. Upper left 3 x 3 matrix is the rotation matrix. Upper right 3 x 1 vector is the position vector, lower 1 x 3 row vector is the perspective transformation and last diagonal element is the scale factor

3.1.2 THE LINK AND JOINT PARAMETERS

In a mechanical manipulator each joint-link pair constitutes a degree of freedom. Thus if an N degree of freedom manipulator is considered there will be N joint - link pairs. Link zero which is not considered as a part of the robot is attached to supporting base. Last link is attached with a tool. Joints and links are numbered outwardly from the base; thus first joint connects first link and supporting base. No closed loops are usually formed. Only lower pairs are used to form joints.

Different parameters which kinematically describe the link - joint pair are established as follows. If a joint axis i is considered [Fig.3.1], it has two normals associated with it, one for each of the links. The relative position of two such connected links, i.e. link i-1 and link i, is given by d_i which is the distance measured along the joint axis between normals. The joint angle θ_i between the normals is measured in a plane normal to the joint axis perpendicular to a_i.Links can be characterized by two parameters a_i and α_i. a_i is the shortest distance measured along the common normal between the joint axis. α_i is the angle between the joint axes measured in a plane perpendicular to a_i. Thus a_i and α_i are the length and the twist angle of the link i. Thus θ_i, α_i, a_i and d_i constitute a sufficient

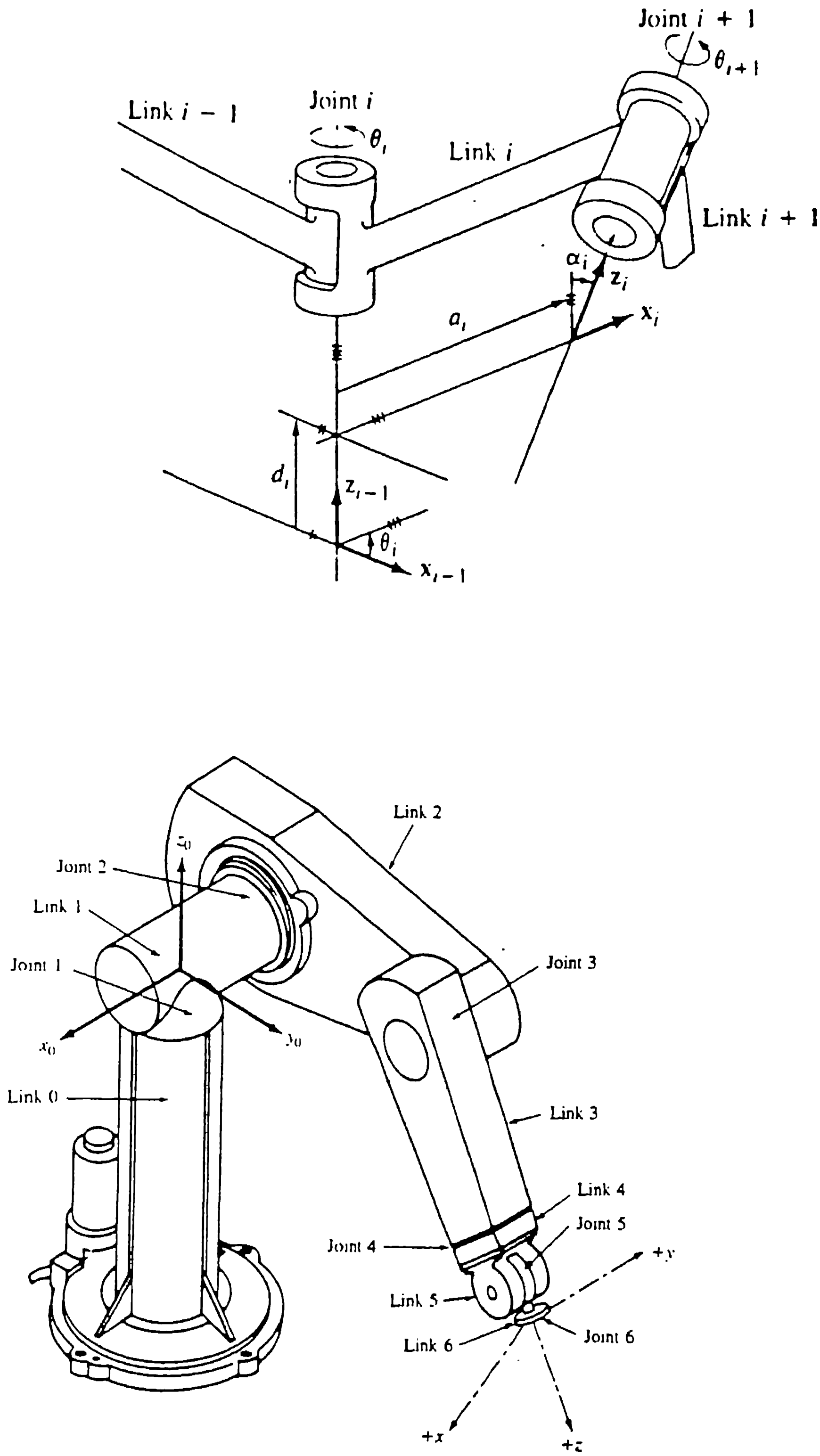


Fig 3.1 Link and joint parameters

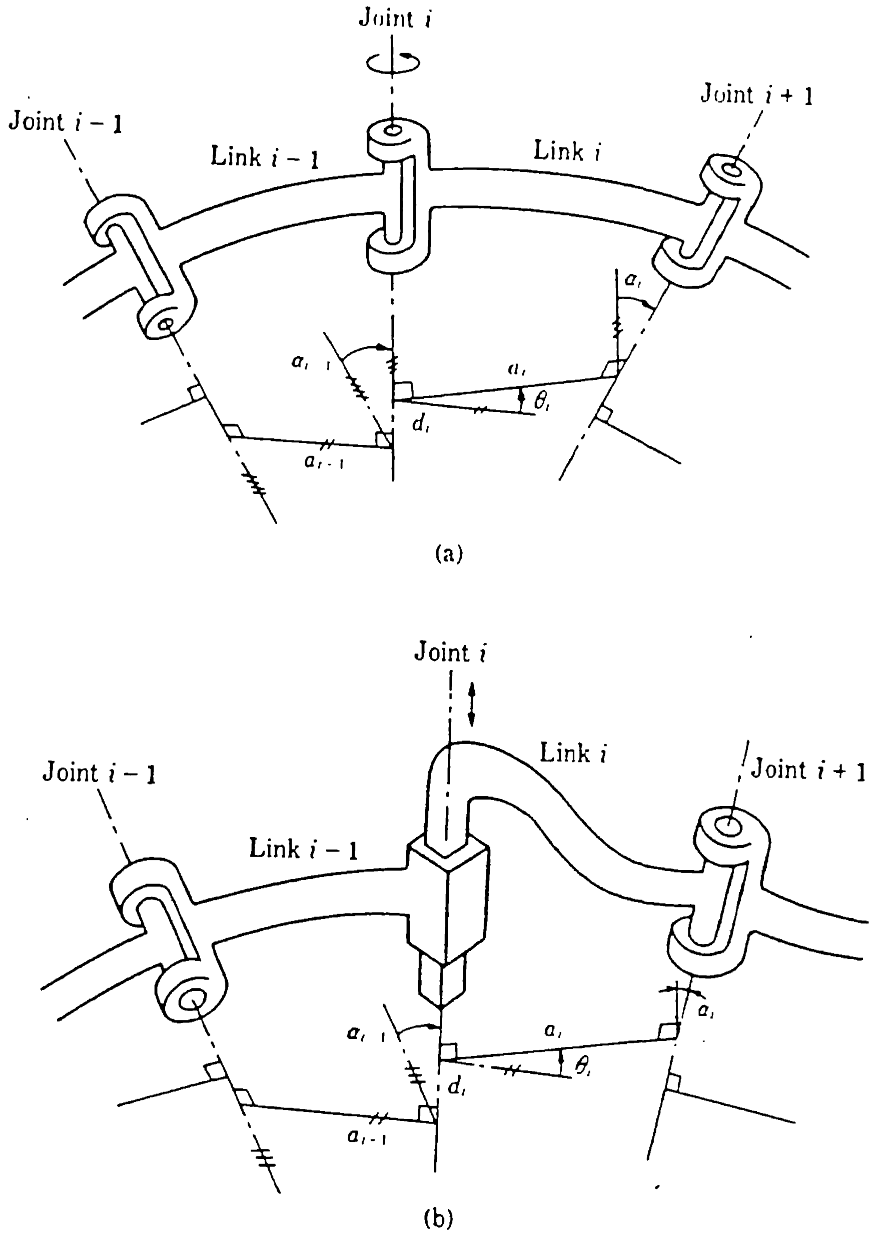


Fig 3.2 Joint axes, joint variables and link parameters

(a) joint i is revolute (b) joint i is prismatic

set to determine completely the kinematic configuration of each link of a robot arm.

3.1.3 THE DENAVIT - HARTENBERG REPRESENTATION

To describe the translational and rotational relationship between consecutive links, Denavit and Hartenberg proposed a matrix method. This representation results in a 4x4 homogeneous transformation matrix representing each link's coordinate system at the joint with respect to the previous link's coordinate system. Coordinate frames are established based on following rules:

1. The Z_{i-1} axis lies along the axis of motion of the i^{th} joint.

2. The X_i axis is normal to the Z_{i-1} axis and pointing away from it.

3. The Y_i axis completes the right handed coordinate system as required.

The D-H representation of a rigid link depends on four geometric parameters associated with each link [Fig. 3.2] Thus

a) θ_i is the joint angle from the X_{i-1} axis to X_i axis about Z_{i-1} axis using right-hand rule.

b) d_i is the distance from the origin of the $(i-1)^{th}$ coordinate frame to the intersection of the Z_{i-1} axis with X_i axis along the Z_{i-1} axis.

c) a_i is the offset distance from the intersection of the Z_{i-1} axis with the X_i axis to the origin of the i^{th} frame along the X_i axis.

d) α_i is the offset angle from the Z_{i-1} axis to the Z_i axis

about X_i axis.

With these specifications the i^{th} frame can be represented in $(i-1)^{th}$ frame as

$$^{i-1}[A]^{i} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i.\cos\alpha_i & \sin\theta_i.\sin\alpha_i & a_i.\cos\theta_i \\ \sin\theta_i & \cos\theta_i.\cos\alpha_i & -\cos\theta_i.\sin\alpha_i & a_i.\sin\theta_i \\ 0.0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \quad (3.1)$$

If $_o\{\mathcal{X}\}^E$ represents the position (x, y, z) of the end-effector in the base frame then

$$_o\{\hat{\mathcal{X}}\}^E = {}_o[A]^1.{}_1[A]^2 \ldots {}_{n-1}[A]^n \; .{}_n[T]^E\{U\} \quad (3.2)$$

where $_{(n)}[T]^E$ is the Tool Transformation and $\{U\}$ is given as

$$\{U\} = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \\ 1.0 \end{bmatrix} \quad (3.2.1)$$

3.2 THE JACOBIAN MATRIX

Classically the Jacobian is a matrix relating differentials in one reference frame to the differentials in another reference frame. In robotics applications, one is interested in relating the cartesian space dx, dy, dz to the differentials in the parameter space; $d\alpha_i$, $d\theta_i$, da_i, dd_i. The Jacobian matrix is formed as collection of partial derivatives of a function of multiple variables. Thus collection of derivative of a vector function of vectors will lead to the formation of a Jacobian. For example the end-effector position vector $\hat{\mathcal{X}}$ is a vector function of vectors $\hat{\theta}$, $\hat{\alpha}$, $\hat{a}$ and $\hat{d}$. In other words

$$\hat{\mathcal{X}} = (\ X\ Y\ Z)^T \quad (3.3)$$

$$X = f_1(\ \hat{\theta},\ \hat{\alpha},\ \hat{a},\ \hat{d}\ ) \quad (3.3.1)$$

$$Y = f_2(\hat{\theta}, \hat{\alpha}, \hat{a}, \hat{d}) \tag{3.3.2}$$

$$Z = f_3(\hat{\theta}, \hat{\alpha}, \hat{a}, \hat{d}) \tag{3.3.3}$$

where

$$\hat{\theta} = (\theta_1, \theta_2, \ldots\ldots\theta_n)^T \tag{3.3.4}$$

$$\hat{\alpha} = (\alpha_1, \alpha_2, \ldots\ldots\alpha_n)^T \tag{3.3.5}$$

$$\hat{a} = (a_1, a_2, \ldots\ldots a_n)^T \tag{3.3.6}$$

$$\hat{d} = (d_1, d_2, \ldots\ldots d_n)^T \tag{3.3.7}$$

thus

$$\hat{\mathcal{X}} = \hat{\mathcal{F}}(\hat{\phi}) \tag{3.4}$$

where

$$\hat{\phi} = (\hat{\theta}\ \hat{\alpha}\ \hat{a}\ \hat{d}) \tag{3.4.1}$$

The differential of $\hat{\mathcal{X}}$ can be written as

$$\mathring{\hat{\mathcal{X}}} = \frac{\partial \hat{\mathcal{F}}}{\partial \hat{\phi}} . \mathring{\hat{\phi}} \tag{3.5}$$

Differential matrix term $\partial\hat{\mathcal{F}}/\partial\hat{\phi}$ is called Jacobian

For any given configuration of the manipulator, joint rates $\dot{\phi}$ are related to velocity of the end-effector in a linear fashion through Jacobian. However this is only an instantaneous relationship, as in the next instant Jacobian has changed slightly.

There are several methods of evaluating the Jacobian. One may be to directly differentiate the forward kinematic equations. This method is tedious and computationally unsuitable. An alternate way is to use the screw coordinate notations for evaluating end-effector Jacobian. Complete derivation of Jacobian using the screw coordinate notation is given in Appendix II.

3.3 TAYLOR SERIES EXPANSION

From the forward kinematic equation we know that the end-effector position vector $\hat{x}$ is a vector function of vectors. Each component of vector $\hat{x}$ is a function of link and joint parameters. All these components can be expanded in form of Taylor series. Thus if $\Delta\hat{\phi}$ is in a small region around $\hat{\phi}$, the expansion of $X(\hat{\phi})$, $Y(\hat{\phi})$, and $Z(\hat{\phi})$ can be written as

$$X(\hat{\phi} + \Delta\hat{\phi}) = X(\hat{\phi}) + \{J_1(\hat{\phi})\}.\Delta\hat{\phi} + \frac{1}{2!}\Delta\hat{\phi}^T.H_x(\hat{\phi}).\Delta\hat{\phi} + \ldots \quad (3.6)$$

$$Y(\hat{\phi} + \Delta\hat{\phi}) = Y(\hat{\phi}) + \{J_2(\hat{\phi})\}.\Delta\hat{\phi} + \frac{1}{2!}\Delta\hat{\phi}^T.H_y(\hat{\phi}).\Delta\hat{\phi} + \ldots \quad (3.7)$$

$$Z(\hat{\phi} + \Delta\hat{\phi}) = Z(\hat{\phi}) + \{J_3(\hat{\phi})\}.\Delta\hat{\phi} + \frac{1}{2!}\Delta\hat{\phi}^T.H_z(\hat{\phi}).\Delta\hat{\phi} + \ldots \quad (3.8)$$

where J_1, J_2, J_3, are the first, second and third rows of the Jacobian matrix. H_x, H_y, H_z, are the Hessian matrices. A Hessian matrix is a collection of second order differential terms. If $\Delta\phi$ is small then second and higher order terms can be neglected and equations 3.6, 3.7 and 3.8 can be combined together as

$$\hat{x}(\hat{\phi} + \Delta\hat{\phi}) = x(\hat{\phi}) + J(\hat{\phi}).\Delta\hat{\phi} \quad (3.8)$$

3.4 ERROR MODEL

Above analysis can be used to develop an error model which will correlate the error at end-effector position with errors on parameter. The difference in end -effector position as calculated using the forward kinematic model and the actual position is present because the actual parameters $\hat{\theta}_a$, $\hat{\alpha}_a$, $\hat{a}_a$ and $\hat{d}_a$ are different from their nominal values $\hat{\theta}_o$, $\hat{\alpha}_o$, $\hat{a}_o$ and $\hat{d}_o$. Thus a small perturbation in parameters in joint space leads to a significant change in the end -effector position in cartesian

space. The objective of the error model is to estimate the actual parameters given their nominal values using the error on end-effector position.

The error in end-effector position and parameters are related by

$$\delta\hat{\mathcal{X}} = \frac{\partial\hat{\mathcal{F}}}{\partial\hat{\theta}}.\Delta\hat{\theta} + \frac{\partial\hat{\mathcal{F}}}{\partial\hat{\alpha}}.\Delta\hat{\alpha} + \frac{\partial\hat{\mathcal{F}}}{\partial\hat{a}}.\Delta\hat{a} + \frac{\partial\hat{\mathcal{F}}}{\partial\hat{d}}.\Delta\hat{d} \qquad (3.9)$$

where

$$\frac{\partial\mathcal{F}}{\partial\hat{\theta}} = J_{\theta} \qquad (3.9.1)$$

$$\frac{\partial\mathcal{F}}{\partial\hat{\alpha}} = J_{\alpha} \qquad (3.9.2)$$

$$\frac{\partial\mathcal{F}}{\partial\hat{a}} = J_{a} \qquad (3.9.3)$$

$$\frac{\partial\mathcal{F}}{\partial\hat{d}} = J_{d} \qquad (3.9.4)$$

J_θ, J_α, J_a and J_d are the Jacobians with respect to θ, α, a and d respectively.Hence equation (3.9) can be written as follows

$$\delta\hat{\mathcal{X}} = [\, J_\theta \; J_\alpha \; J_a \; J_d] \begin{bmatrix} \Delta\hat{\theta} \\ \Delta\hat{\alpha} \\ \Delta\hat{a} \\ \Delta\hat{d} \end{bmatrix} \qquad (3.10)$$

If D be the matrix of all the Jacobians and $\Delta\hat{\phi}$ be the vector containing all the parameter error vectors i,e,

$$[D] = [\, J_\theta \; J_\alpha \; J_a \; J_d] \qquad (3.10.1)$$

$$\Delta\hat{\phi} = [\Delta\hat{\theta} \; \Delta\hat{\alpha} \; \Delta\hat{a} \; \Delta\hat{d}]^T \qquad (3.10.2)$$

Equation (3.10) can be rewritten as

$$\delta\hat{x} = [D]\{\Delta\hat{\phi}\} \qquad (3.11)$$

3.4.1. The Minimum Norm Solution

Let the measured value of end-effector position be $\hat{x}_m$. If we assume that measurement errors are negligible then we can say

$$\delta\hat{x} = \hat{x}_m - \hat{x}_c \qquad (3.12)$$

where $\hat{x}_m$ is the measured end-effector position and $\hat{x}_c$ is the end-effector position as given by the kinematic model. We assume that the error on parameters remains constant over entire work space,

which is true when the joint and links are rigid.

$$\begin{bmatrix} \delta x_1 \\ \delta x_2 \\ \vdots \\ \delta x_p \end{bmatrix} = \begin{bmatrix} J_\theta^1 & J_\alpha^1 & J_a^1 & J_d^1 \\ J_\theta^2 & J_\alpha^2 & J_a^2 & J_d^2 \\ \vdots & & & \vdots \\ J_\theta^p & J_\alpha^p & J_a^p & J_d^p \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta\alpha \\ \Delta a \\ \Delta d \end{bmatrix} \qquad 3.13$$

Let $\delta\hat{R}$ be collection of different error vectors

$$[\delta\hat{R}] = [\delta\hat{x}_1 \; \delta\hat{x}_2 \; \delta\hat{x}_3 \; . \; . \; . \; \delta\hat{x}_p]^T \qquad (3.14)$$

and

$$[C] = [D_1 \; D_2 \; . \; . \; . \quad D_p]^T \qquad (3.15)$$

D_1, D_2, ... are as given in equation 3.10.1. Equation 3.13 can be written as

$$[\delta R] = [C]\{\Delta\hat{\phi}\} \qquad (3.16)$$

where $\Delta\hat{\phi} = \phi - \phi_o$ is the error in the total parameter error set.

ϕ_o is the current estimate and ϕ is the corrected estimate provided by minimizing ψ ; ψ is given as

$$\psi = (\delta\hat{\mathcal{R}} - C\Delta\hat{\phi})^T(\delta\hat{\mathcal{R}} - C\Delta\hat{\phi}) \tag{3.17}$$

this optimization yields

$$\Delta\hat{\phi} = [C\ C^T]^{-1}[C]^T\delta\hat{\mathcal{R}} \tag{3.18}$$

The expression $[C\ C^T]^{-1}[C]^T$ is called as pseudo-inverse of C. The derivation of equation 3.18 and some other properties of pseudo-inverse are given in Appendix III.

3.4.2. Iterative Updating

Since the relation between end-effector position and geometrical parameters is highly non-linear to establish the real parameters it is necessary to iteratively update parameters and recalculate the end-effector position error until the error vanishes. In a real system this may not become zero but converge to some small values.

The iterative updating procedure can be written in pseudo code as

Start

$i=1$

$\delta\hat{R}_o = \hat{F}\ (\hat{\phi}_o)$;

$\delta\hat{R}(i) = \hat{R}_m - \hat{R}_c(i-1)$;

$\Delta\hat{\phi}(i) = (C^TC)^{-1}\ C^T\ \delta\hat{R}(i)$;

$\hat{\phi}(i) = \hat{\phi}(i-1) + \Delta\hat{\phi}(i)$;

$\hat{R}_c(i)= \hat{F}\{\hat{\phi}(i)\}$;

if $[\delta\hat{R}]^T\ [\delta\hat{R}] > 0$ goto start

end;

3.5 DISCUSSION

The goal of a kinematic identification algorithm is to identify the parameters of a kinematic model which describe the actual position and orientation of the end-effector in terms of the measured joint positions, and incorporate the geometrical variations in the structure caused by manufacturing errors.

In this chapter we have presented an efficient procedure for kinematic calibration, based on the Denavit-Hartenberg link parameters. The procedure involves iteration of the linearized kinematic equations. One important aspect of this linearization is the use of efficient vector methods to evaluate the various Jacobians. Orientation errors have been neglected in the present work as no accurate measuring device was available to measure the orientations.

CHAPTER 4

Implementation of Algorithm and Numerical Verification

The Newton-Raphson algorithm described in previous chapter has been implemented on the HP 9000 machines. The programming language used was C. The program is modular and uses around 12 different routines. A separate program was written to generate data for numerical verification. The program provides the flexibility of testing for different robot with varying number of degrees of freedom. The algorithm was tested for 3 different types of robots.

4.1 PROGRAM FOR PARAMETER ESTIMATION

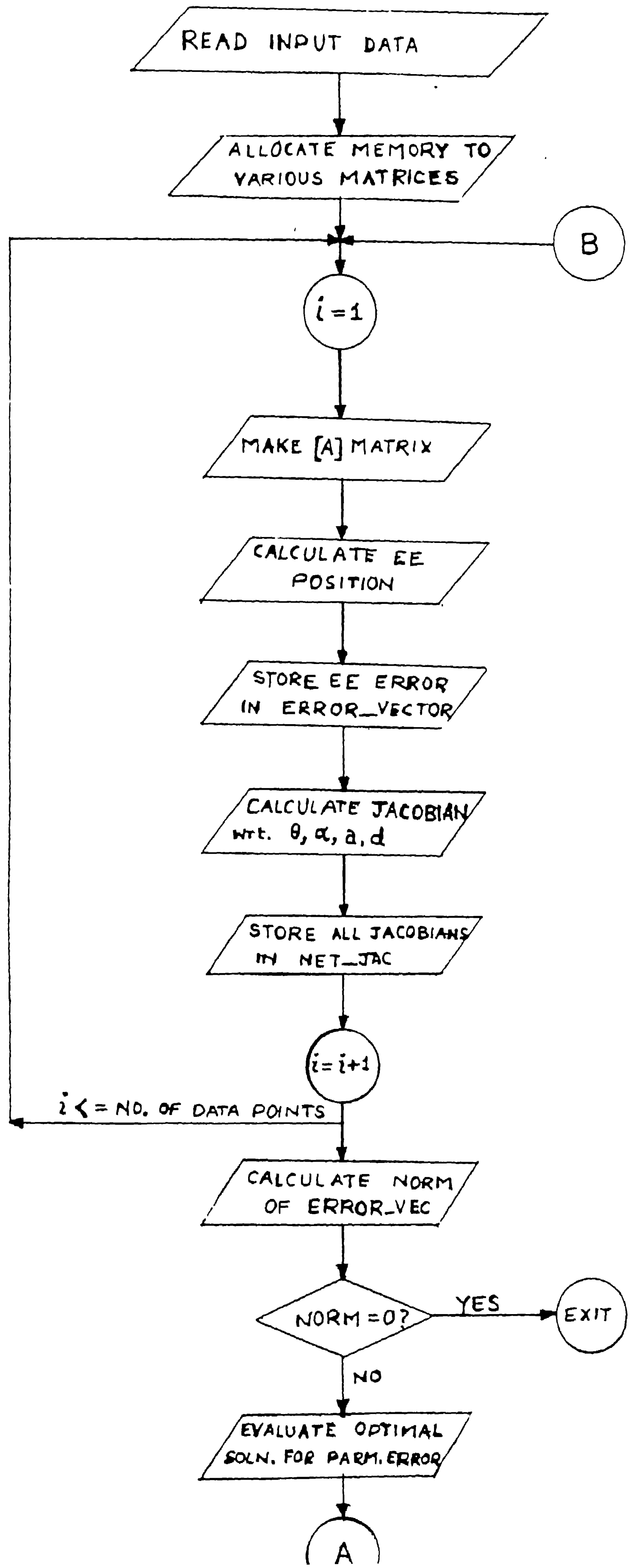
Figure 4.1 shows the program flow. Basic steps in the computation can be listed as follows

a. Read input data which contains, the Denavit-Hartenberg constant parameters of the robot to be calibrated, the joint variables and the corresponding measured (X Y Z) positions of end-effector, for different points in the work-space. All these data points are stored in an array. This task is accompanied by a function called *meas_EE_pos*.

b. Repeat the steps (i) to (iv) for all the data points.

(i) Give a function call to *make_A_mat*. This function makes A matrices for current joint variables.

(ii). Corresponding to current A matrices and the tool-transform, end-effector position is computed by using the forward kinematics routine *calc_EE-pos*.


READ INPUT DATA
ALLOCATE MEMORY TO VARIOUS MATRICES
B
i = 1
MAKE [A] MATRIX
CALCULATE EE POSITION
STORE EE ERROR IN ERROR_VECTOR
CALCULATE JACOBIAN wrt. θ, α, a, d
STORE ALL JACOBIANS IN NET_JAC
i = i+1
i <= NO. OF DATA POINTS
CALCULATE NORM OF ERROR_VEC
NORM = 0?
YES
EXIT
NO
EVALUATE OPTIMAL SOLN. FOR PARM. ERROR
A

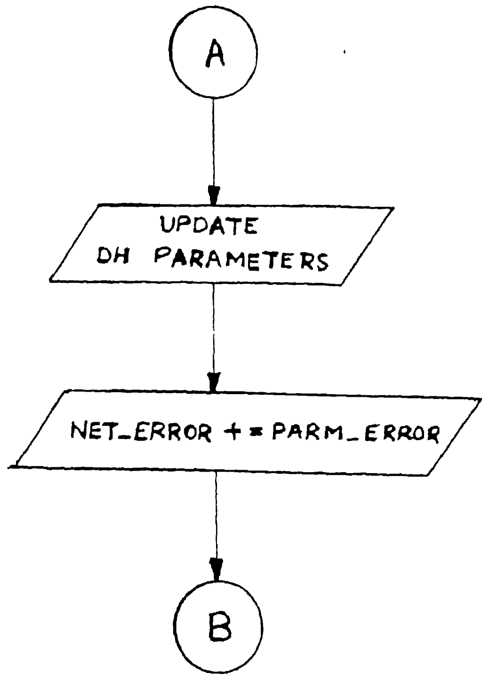


Fig. 4.1 Flow chart of the program for parameter estimation.

(iii). The error in calculated value of end-effector position and measured position of end-effector is evaluated and stored in an array called error_vec.

(iv) Corresponding to current values of A matrices the Jacobian matrices are formed. This task of making Jacobian matrices is accomplished by function *make_jacobian.* This function in turn gives call to functions for forming jacobian with respect to theta, alpha, the link lengths and the link offsets. These functions are called *jacob_theta*, *jacob_alpha*, *jacob_AA* and *jacob_DD* respectively.

(v) All these jacobians are stacked together in a big array called *NET_JAC*.

C. *Error_vec* contains error on the end-effector position for all data points and *NET_JAC* contains corresponding jacobians. These arrays are then passed to function *opt_sol*. This function calls NAG routine f04jgf which gives the optimal solution for a set of equations where number of equations are larger than number of variables. In other words *opt_sol* evaluates the optimal solution for the parameter errors. This error in parameters is stored in an array *parm_err*.

d. These errors on parameters are added to the nominal parameters. This process of updating of D-H parameters for all the data points is done by function *update_DH*.

e. The array *error_vec* is send to function *norm* which will evaluate the euclidean norm of *error_vec*.

f. If this norm is greater than zero, steps b to f are repeated.

4.2 FUNCTIONS

Following functions are used to back the main program.

(i) make_A_mat

This function makes the A matrices corresponding to current data point. Addresses of these A matrices are stored in a global array *A_MAT_ADD*.

(ii) mul2

It is a general purpose matrix multiplication routine. Matrix having different dimensions can be multiplied using this routine.

(iii) calc-EE_pos

The end-effector position corresponding to current value of A matrices is evaluated by this function.

(iv) aalloc

An important feature of the program is dynamic memory allocation. Most of the arrays are not predeclared but memory is allocated to them depending upon the requirements. Sizes of these arrays are dependent on number of data points or degrees of freedom of the robot. This function calls the C library function *calloc* for memory allocation.

(v) make-jacobian

This function evaluates the jacobian matrices with respect to theta, alpha, the link length and the link-offset. This evaluation is done by using the screw notations. Evaluated jacobians are stored in global arrays. *J_theta*, *J_alpha*, *J_AA* and *J_DD*.

(vi) meas-EE-pos

Function opens the input data file and reads the data and stores it in an array. The name of input file is interactively given to the program.

(vii) opt-sol

The optimal solution for the parameter error is determined using this function. This function calls the NAG library routine *fo4jgf* to solve the matrix equation. In other words it evaluates the pseudo inverse of *NET_JAC*.

(viii) update_DH

This routine merely updates all the D-H parameters corresponding to all the data points.

(ix) norm

The function evaluates the euclidean norm of any vector supplied to it.

(x) printmat

This function was written for the purpose of debugging. It is a useful function can print matrix of any dimension on the standard output.

4.3 NUMERICAL VERIFICATION THROUGH EXAMPLES

For the purpose of verification of the algorithm it was tested for three different robots having different number of degrees of freedom. For simulation it was assumed that the measured end effector position is known. The input data was generated by using a separate program. To get the perturbed value of end effector position a known set of parameter errors is added

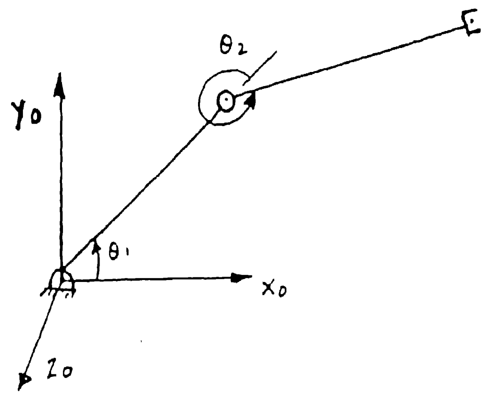


Fig 4.2 Kinematic diagram of robot as example 1.

Joint i	θ_i °	α_i °	a_i mm	d_i mm
1.	θ_1	0.0	1000.0	0.0
2.	θ_2	0.0	1000.0	0.0

Table 4.1 : Denvit-Hartenberg parameters for robot shown in Fig. 4.1.

Joint i	$\delta\theta_i$ °	$\delta\alpha_i$ °	δa_i mm	δd_i mm
1.	4.0	3.0	6.0	0.0
2.	2.0	0.0	10.0	0.0

Table 4.2 : Parameter errors chosen for example 1.

Joint variables		End-effector position with parm. errors		
θ_1°	θ_2°	X (mm)	Y (mm)	Z (mm)
30.0	30.0	1245.22	1484.62	28.01
32.0	32.0	1159.76	1539.77	29.55
35.0	36.0	1009.54	1616.54	32.54
38.0	39.0	871.29	1674.94	34.67
40.0	40.0	794.75	1705.70	35.37
42.0	42.0	699.51	1732.98	36.71
45.0	45.0	555.18	1763.10	38.65
47.0	47.0	458.52	1775.80	39.89
48.0	49.0	393.00	1776.19	41.07
50.0	50.0	313.80	1784.10	41.65

Table 4.3 : End-effector position with erroneous D-H parameters for different joint variables

Iteration No.	Norm of error (Sq. meter)	Average errors (meters)		
		X	Y	Z
1.	0.281682	-0.1432	0.0736	0.0358
2.	0.000657	0.0023	0.0074	-0.0002
3.	0.000001	-0.0000	0.0000	0.0000
4.	0.000000	0.0000	0.0000	0.0000

Table 4.4 : Convergence of the algorithm for two d.o.f. case.

Joint i	$\delta\theta_i$°	$\delta\alpha_i$°	δa_i mm	δd_i mm
1	3.9998	2.9999	6.0011	0.05754
2	2.0002	0.0000	9.9990	-0.0574

Table 4.5 : Parameter errors identified using the algorithm.

to nominal parameter values.Then using the forward kinematics routine with perturbed parameters values, the end-effector position are calculated. This generated value of end effector position is fed as input along with the corresponding joint angle variables and Denavit-Hartenberg parameters to the parameter estimation program. This program then estimates the parameter errors. The parameter errors obtained from this algorithm must be the same as fed initially to input data generation program.

This verification was done for three different cases as described in subsequent sections. The various observations are also discussed.

4.3.1 Example 1

Fig. 4.2 shows a simple two degrees of freedom planer manipulator. The corresponding nominal value of Denavit-Hartenberg parameters are listed in table 4.1. Assuming that the actual parameters differ from the nominal parameters and their difference is as given in table 4.2. Using the program for data generation ten different (X Y Z) positions of end effector are obtained for ten sets of joint variables. These ten sets of variables and the corresponding end-effector positions are shown in table 4.3. Using the data in table 4.3 as input to the parameters estimation algorithm the errors on different parameters is estimated. Table 4.4 shows that the algorithm converges in 3 iterations and the estimated parameters errors as shown in table 4.5 are exactly same as in table 4.2, which were given as

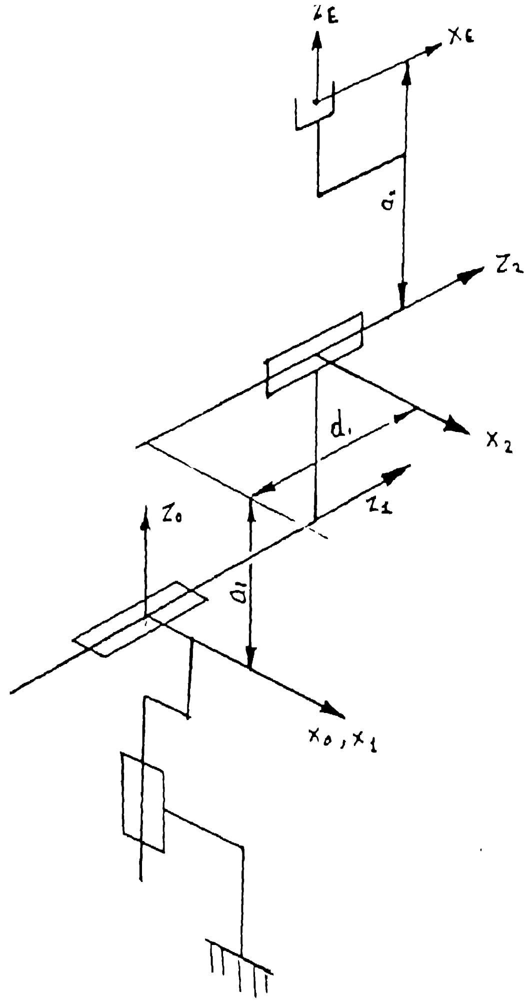


Fig 4.3 Kinematic diagram of robot as example 2.

Joint i	θ_i	α_i	a_i	d_i
1.	θ_1	-90.0	0.00	0.00
2.	θ_2	0.0	235.0	75.0
3.	θ_3	0.0	75.0	0.0

Table 4.6 : Denavit-Harteberg parameters for robot shown in Fig. 4.3

Joint i	$\delta\theta_i$ °	$\delta\alpha_i$ °	δa_i mm	δd_i mm
1	4.0	3.0	6.0	4.0
2	2.0	0.0	10.0	0.0
3	5.0	5.0	7.0	0.0

Table 4.7 : Errors on the D-H parameters.

Joint variables			End-effector position with parm.errors		
θ_1	θ_2	θ_3	X	Y	Z
30.0	-30.0	-5.0	233.59	262.70	93.30
32.0	-32.0	5.0	216.75	265.79	84.59
35.0	-36.0	10.0	195.32	270.08	97.11
38.0	-39.0	15.0	174.40	273.61	103.63
40.0	-40.0	20.0	160.10	275.34	99.41
42.0	-42.0	25.0	144.55	274.63	100.51
45.0	-45.0	30.0	123.19	273.53	106.50
47.0	-47.0	35.0	107.40	270.25	107.40
48.0	-49.0	40.0	96.79	264.20	108.28
50.0	50.0	45.0	82.27	260.52	104.75

Table 4.8 : End-effector positions with errooneous parameters.

Ieration Number	Rank of Jacobian	Norm of error (Sq. meter)	Average errors (meters)		
			X	Y	Z
1.	10	0.006603	-0.004458	0.024913	-0.00139
2.	10	0.000046	0.002086	-0.000285	0.00027
3.	10	0.000000	-0.000001	0.000014	0.00000
4.	10	0.000000	0.000000	0.000000	0.00000

Table 4.9 : Convergence results for three d.o.f. case.

Joint i	$\delta\theta_i$ °	$\delta\alpha_i$ °	δa_i mm	δd_i mm
1	4.0001	3.0007	6.001	3.991
2	1.9990	0.0000	10.002	**
3	4.9985	5.0008	6.997	**

Table 4.10 : Estimated parameter errors.

initial perturbation.

4.3.1.1 Observation

1. It is found that the convergence of algorithm for this case is independent of the choice of data points in the workspace.

2. Since the consecutive joint axes are parallel the jacobian with respect to link offset d is singular hence error on these are not observable.

3. The largest correction on parameters is obtained in the first step an algorithm shows a very rapid convergence as is evident in table 4.4.

4.3.2 Example 2

A three degrees of freedom robot was chosen for second example. Fig. 4.3 shows the kinematic diagram and table 4.6 shows the nominal Denavit-Hartenberg parameters. Table 4.7 shows the assumed values of parameter errors. Table 4.8 shows the ten different sets of joint variable and the corresponding end-effector position with errors on the Denavit-Hartenberg parameters.

With input as given in table 4.8, the results of parameters estimation algorithm are shown in table 4.9. Estimated parameter errors are shown in table 4.10 which are exactly same as in table 4.7.

4.3.2.1 Observations

1. Unlike previous example it is found that the rank of jacobian matrices with respect to link length and link offset are dependent on choice of data points in work space but this effect

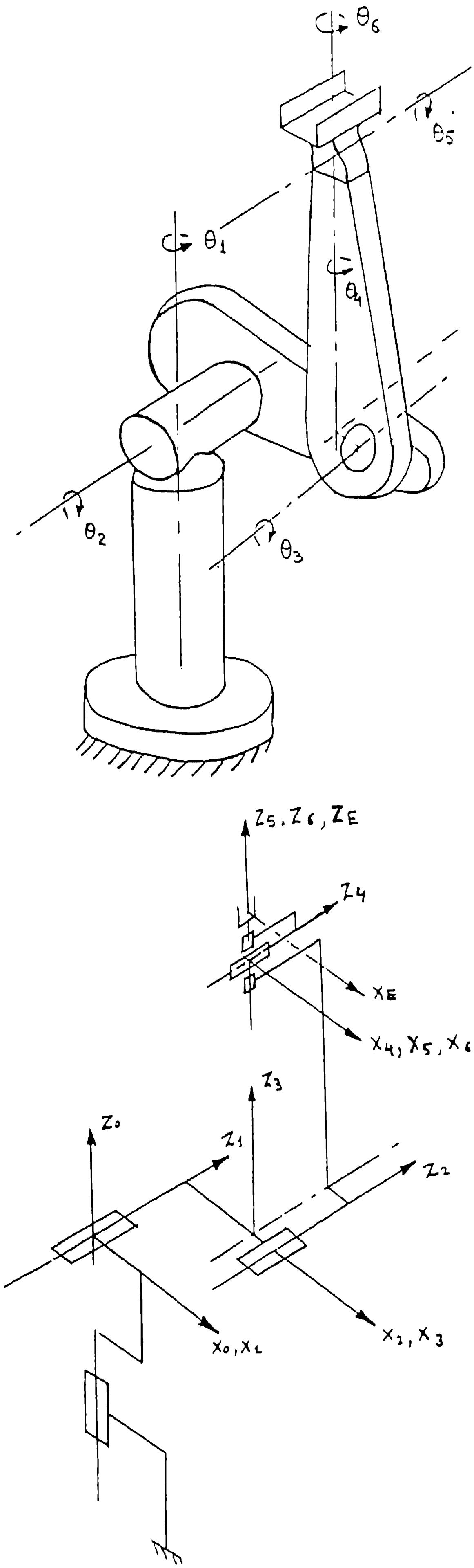


Fig 4.4 PUMA 560 : (a) schematic diagram (b) Kinematic diagram.

Joint i	θ_i °	α_i °	a_i mm	d_i mm
1	θ_1	-90.0	0.0	0.0
2	θ_2	0.0	431.80	149.09
3	θ_3	90.0	-20.32	0.00
4	θ_4	-90.0	0.0	433.07
5	θ_5	90.0	0.0	0.0
6	θ_6	0.0	0.0	0.0

Table 4.11 : The D-H parameters for PUMA 560.

Joint i	$\delta\theta_i$ °	$\delta\alpha_i$ °	δa_i mm	δd_i mm
1	3.0	5.0	5.0	0.0
2	4.0	0.0	10.0	0.0
3	2.0	-3.0	-3.0	0.0
4	4.0	-5.0	0.0	0.0
5	5.0	2.0	0.0	0.0
6	4.0	0.0	6.0	0.0

Table 4.12 : Errors selected for various parameters.

Joint Variables						End-effector position with parameter errors		
θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	X	Y	Z
30.00	-5.00	35.0	-5.0	5.0	60.0	502.12	598.43	485.32
35.0	-10.0	45.0	5.0	15.0	50.0	480.41	673.66	478.41
40.0	-15.0	55.0	10.0	25.0	40.0	438.47	743.23	456.99
45.0	-20.0	65.0	15.0	35.0	30.0	374.25	795.45	430.55
50.0	-25.0	75.0	20.0	45.0	20.0	293.80	825.07	403.93
55.0	-30.0	85.0	25.0	55.0	10.0	205.23	830.03	381.19
60.0	-35.0	95.0	30.0	65.0	0.0	117.02	811.71	364.86
65.0	-40.0	105.0	35.0	75.0	-10.0	36.45	774.42	355.67
110.0	-45.0	-5.0	40.0	85.0	-20.0	-254.88	34.26	589.91
120.0	50.0	-10.0	45.0	95.0	-30.0	-506.20	292.58	-137.65

Table 4.13 : End-effector positions for various joint variables

Ieration number	Rank of Jacobian	Norm of error (Sq. meter)	Average errors (meters)		
			X	Y	Z
1	18	0.046451	-0.01180	0.03483	-0.04074
2	18	0.000648	0.00576	0.00227	-0.00141
3	18	0.000001	-0.00021	0.00004	-0.00008
4	18	0.000000	0.00000	-0.00000	0.00000

Table 4.14 : Convergence results for six d.o.f. case.

Joint i	$\delta\theta_i\,^\circ$	$\delta\alpha_i\,^\circ$	δa_i mm	δd_i mm
1	3.0000	4.9999	5.0032	* *
2	4.0096	0.0000	9.9996	* *
3	2.0007	-2.9999	-3.0025	* *
4	3.9992	-5.0016	0.0000	* *
5	5.0001	2.0015	0.0000	* *
6	4.0096	0.0000	5.9993	* *

Table 4.15 : Estimated errors on various parameters for PUMA.

can be nullified by changing the tolerance of NAG routine ***f04jgf*** to a slightly lower value. This routine besides solving the matrix equation also gives the rank of jacobian matrix.

2. Convergence in this case again is rapid and norm of error reduces to zero in three iterations.

3. The parameter errors on link offsets corresponding to parallel joint axes are again unobservable for the reason stated earlier, that its jacobian is singular. But error on link offset with non parallel axes is observable as shown in table 4.10. The tenth entry is the error on link offset for non-parallel axis.

4.3.3 Example 3

The third example is a six degrees of freedom PUMA 560 robot. Figure 4.4 shows the PUMA 560 and its kinematic diagram. Table 4.11 contains the nominal Denavit-Hartenberg parameter values of the robot. Table 4.12 shows the assumed parameter errors. Table 4.13 contains ten sets of joint variables corresponding to which different positions of end-effector are evaluated.Table 4.13 form the input to program for parameter estimation. The convergence for this case is also rapid as shown in table 4.14. The estimated parameters are shown in table 4.15.

It is essential to monitor the rank of ***NET_JAC*** which is the matrix formed by stacking jacobians with respect to different parameters. If this rank is less than number of parameters, then algorithm will not converge to the values which were initially given as perturbations. The task of maintaining the rank equal to number of parameters is accomplished by selecting

appropriate input joint variables, suitably selecting the tolerance or accuracy for NAG routine and avoiding the work space singularities.

The error on d parameter i.e. the link offset has been neglected as jacobian with respect to this parameter is a singular matrix. Hence the results presented are pertaining to 18 parameters, six theta values, six alpha values and six length parameters.

4.3.3.1 Observations

1. The rank of jacobians with respect to link length and link offset is highly dependent on the choice of joint variables. If joint variables are moved in small steps of one or two degrees it is found that the NAG routine is unable to identify the small change in jacobian entries and declares them to be singular. Though they actually may not be singular. To avoid this the step size is chosen to be five to ten degrees.

2. None of the joint should be fixed to constant value. This practically reduces the degree of freedom and one of the parameter set may become unobservable. Though this may not happen for all the regions in work space.

3. The input variables must be maintained so as to avoid work space and boundary singularities. To avoid work space singularity θ_4 must not be fixed. Boundary singularities can be avoided by maintaining the end-effector inside the work space boundary.

4. The stacking of jacobians with respect to theta, alpha, link length a and link offset d causes the problem of so called *data diversity*. The jacobians with respect to theta and alpha have

entries with large magnitude and those with respect to link length and link offset have small magnitude. This causes a typical problem to the NAG routine for optimal solution. With this kind of diversity if tolerance value of the routine is reduced to a lower value say of the order of 10^{-6} then the algorithm loses its convergence or in other words the pseudo-inverse of ***NET_JAC*** becomes unreliable. If the tolerance is increased to the orders of 10^{-4} the routine is unable to distinguish between the slightly different entries in jacobian with respect to link length and link offset. Hence to have a significant difference in them, step size of joint variables should be sufficiently large. One cannot use arbitrarily large value of steps as algorithm demands some sort of trajectory to be traced in work-space.

5. With above conditions met the algorithm indeed shows a rapid convergence to exact values.

4.4 DISCUSSION

The numerical verification of the algorithm, as described above, gives rise to the following issues

(a) Rate of Convergence.

(b) Convergence to the desired solution.

(c) Presence of singularities.

Number of iterations required for the convergence are directly related to the initial perturbation selected. In the numerical verification shows that the algorithm converges within three or four iteration irrespective of number of degrees of freedom. The Jacobians evaluated from the routine using vector method was also

cross checked by making hand calculation for a three degree of freedom case.

The Newton-Raphson algorithm has two practical disadvantages. First it requires the evaluation of the huge inverse Jacobian at each iteration. Inversion of the Jacobian matrix is a major contributor to it´s computational complexity. If the end-effector position is close to being degenerate, the algorithm becomes unstable. The Jacobian matrix becomes singular in this case. These configuration can always be avoided by selecting a proper trajectory. Another factor adding to the complexity of algorithm is generation of various Jacobian matrices.

CHAPTER 5

CALIBRATION OF THE MASTER-SLAVE SYSTEM

The procedure for calibration of a teleoperated system involves two modules. One is the calibration of master arm and other being the calibration of slave arm. It is also important to find out the exact relationship between the joint variables of two robots. The precision considered here concerns that of the actual configuration of slave arm with respect to a configuration requested by the control system. This is manifested in the task space by positional precision of the end-effector and it´s orientation precision.

Present work emphasizes on determining the errors in correspondence of two modules at joint angle level. Initially it was planned to implement the parameter estimation algorithm thereby determining the accurate Denavit-Hartenberg parameters of both the master and the slave robot. Due to unavailability of appropriate measuring device for measuring the end-effector position, the work has been limited to the level I i,e, joint angle calibration. A stepwise procedure has been given to perform the level II calibration or calibration at kinematic parameter level. An attempt to identify the various sources of errors in a tele-operated system is made.

5.1 CALIBRATION OF A MASTER-SLAVE TELE-OPERATED SYSTEM.

5.1.1 A TYPICAL POSITION CONTROL STRATEGY

A master slave tele-operated robot has two working modules, namely the master and the slave robots. Positional and orientation errors creep in both of these module. For controlling the posture of slave in exact correspondence with the master, it is necessary to solve the inverse kinematics for the master and forward kinematics for the slave. A typical strategy to control the slave pose can be laid down as follows

(a) Detemine the joint transducer readings for each end-effector pose throughout the workspace. This is the inverse kinematics problem.

(b) Determine a relationship between joint transducer readings of master and those of slave. If the kinematic equivalence between the two robots has been ensured, then such a relationship must make the joint variables of two exactly same.

(c) Final step will be to find out the end-effector posture of the slave.

This control procedure would be called open loop control. If a high positional accuracy is desired, it is essential to have some feedback signal in term of actual position of the slave. This feedback signal would then be utilized to update the joint transducer readings of the master.

5.1.2 KINEMATIC CALIBRATION PROCEDURE

The need for calibration arises as errors occur at all the steps of position control. Moreover most of the teleoperated system adopt open loop control so it is essential to have calibration.

The inverse kinematics for the master and the forward kinematics for the slave will be affected by the parameter errors as described earlier. The correlation between joint transducer readings may also be erroneous. The kinematic calibration procedure for a master-slave, employing an open loop control will be as follows

(i) Parameter Estimation.

The first step would be to estimate the exact Denavit-Hartenberg parameters for both of the robots. This will involve measurement of the end-effector position along a trajectory. This trejectory should be such as to avoid the singularities. These accurately measured positions and corresponding joint variables can be fed to parameter estimation program as described earlier to determine the correct Denavit-Hartenberg for both the robots.

(ii) Updating the Control Software.

These estimated parameters for entire workspace should be substituted for the nominal values and hence permanently change the control software.

(iii) Measurement of Positional Errors.

To measure the correspondence between the end-effector pose of the master and the slave, two fixture plates with accurately drilled holes in exact corresponding position can be used. Such a procedure determines the errors in certain plane in the workspace. The distance between fixture plates should be same as that between base frame of the two robots. The error in end-effector position of slave can be measured by touching the end-effector of master at a hole on fixture plate and measuring the offset of slave end-effector in X, Y and Z directions.

(iv) These positional errors now can be fed to the Jacobian control routine to determine the correction in joint variables such that the slave would go in exact corresponding position.

(v) This procedure should be repeated for the entire workspace.

In above procedure step (ii) is crucial. When parameters assume values different from the nominal values, it becomes almost impossible to develop a closed form solution for inverse kinematics. Though the solution of the forward kinematics is still not unmanageable. For solving inverse kinematics with updated parameters some numerical technique has to be used. The algorithm for parameter estimation is in fact an inverse kinematic solution with non-nominal parameters. The initial guess of the solution would be provided by the solution of inverse kinematics

with nominal parameter values. Thus control software has to be modified accordingly to accommodate the solution for inverse kinematics with non-nominal parameters.

5.2 CALIBRATION OF MA2000 MASTER-SLAVE SYSTEM.

The procedure described above can be implemented only if an accurate measurement of the end-effector position is available. In absence of such a device only joint angle level calibration can be done. The present work concentrates on the following issues

(i) To determine the relationship between the potentiometer readings and the actual joint variable values. In other words to determine the characteristics of the potentiometers.

(ii) To determine the variation in the joint variable values of two robots.

(iii) To determine the error in end-effector position of two robots due to the errors in joint variables over a plane.To establish a relation between the potentiometer reading and actual joint variables it was necessary to have some angle measuring device. The digitized output voltage value is available from the control program. Joint variable were measured by placing protractors at the joint axes.

5.2.1 CONTROL OF MA2000 MASTER-SLAVE SYSTEM

The control of the master-slave system does not

involve any forward or inverse kinematics. The joint position data available from master robot is placed in the controller of MA200. The controller provides an interface with the host computer through shared memory. Certain number of bits are allocated to different informations. For instance 12 bits of joint position data can be put in this area. Controller checks this area continually and positions the robot in commanded posture.

For operating MA2000 in teleoperated mode the joint position data available from master robot is placed in the controller which then positions the slave in requested posture. The joint position data from master is appropriately scaled before sending them to the controller. The scale factor is the ratio of difference between maximum and minimum potentiometer readings of slave and the master. Joint position data for slave is determined as described below. If

N_c^m = Current pot. reading of master.

N_{min}^m = Minimum pot. reading of master.

N_{max}^m = Maximum pot. reading of master.

N_c^s = Current pot. reading of slave.

N_{min}^s = Minimum pot. reading of slave.

N_{max}^s = Maximum pot. reading of slave.

$$N_c^s = N_{min}^s + (N_c^m - N_{min}^m)\ CAL \qquad (5.1)$$

Where

$$CAL = \frac{N^{s}_{max} - N^{s}_{min}}{N^{m}_{max} - N^{m}_{min}} \qquad (5.1.1)$$

The maximum and minimum value for slave have been determined by trial and error.

5.2.2 ERROR IDENTIFICATION

Figure 5.1, 5.2, 5.3, 5.4, 5.5 and 5.6 show the characteristics of potentiometers at master waist, master shoulder, master elbow and potentiometers at corresponding joints of slave respectively.

Figures 5.7, 5.8 and 5.9 show the desired and actual relation between the joint variables at waist, shoulder and elbow joints respectively.

Such a variation in error is mainly due to erroneous correspondence between the ranges of two robot joints. The minimum and maximum values of joint variables of the master and the slave are not same. The error is present as the range of the slave arm has been decided approximately. Due to this *range shrinking* joint error becomes a function of joint variable of the master. Equation 5.1 can be written in terms of joint variables as follows

$$\theta^{s}_{c} = \theta^{s}_{min} + (\theta^{m}_{c} - \theta^{m}_{min}) * CAL\_\theta \qquad (5.2)$$

where

$$CAL = \frac{\theta^{s}_{max} - \theta^{s}_{min}}{\theta^{m}_{max} - \theta^{m}_{min}} \qquad (5.2.1)$$

POTENTIOMETER CHARACTERISTICS

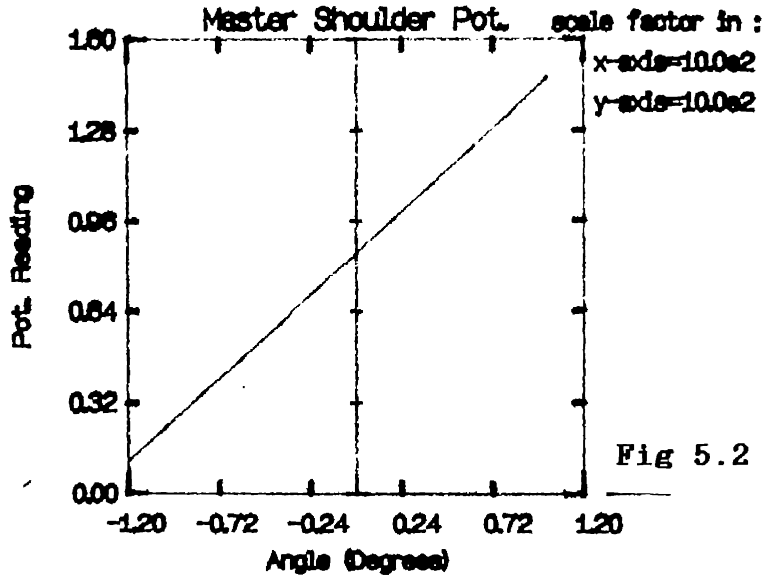


Fig 5.2

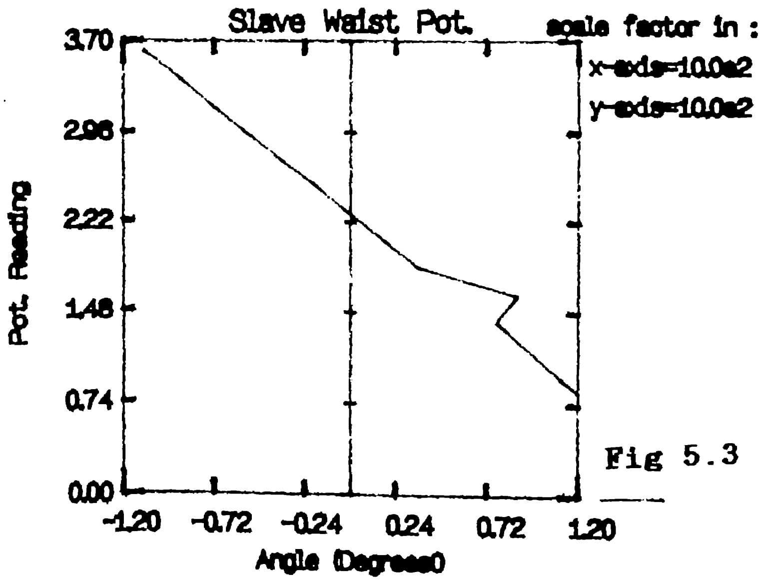


Fig 5.3

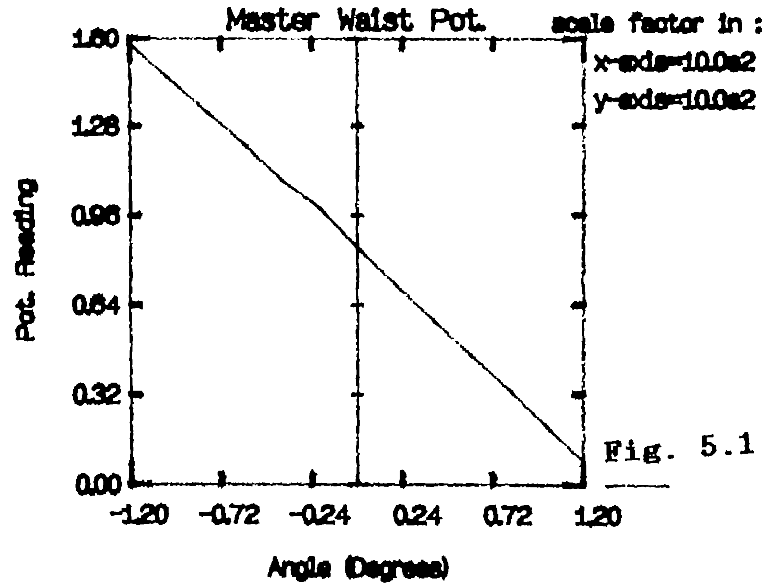


Fig. 5.1

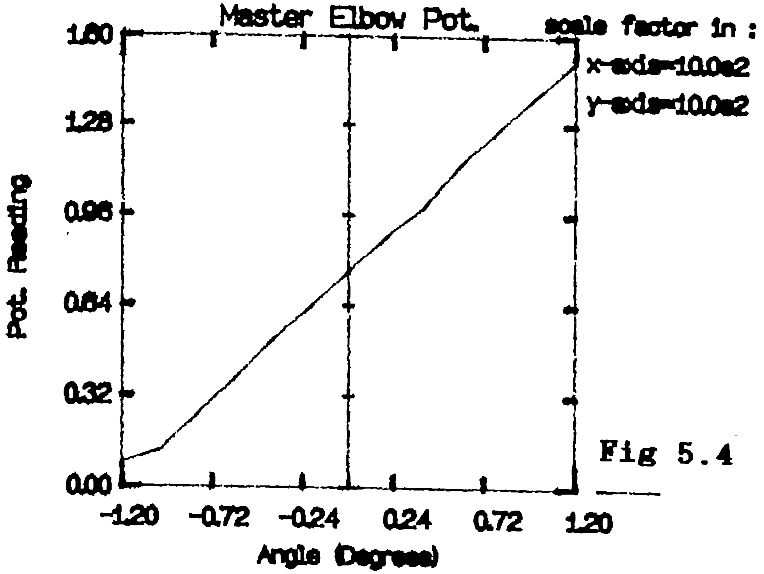


Fig 5.4

POTENTIOMETER CHARACTERISTICS

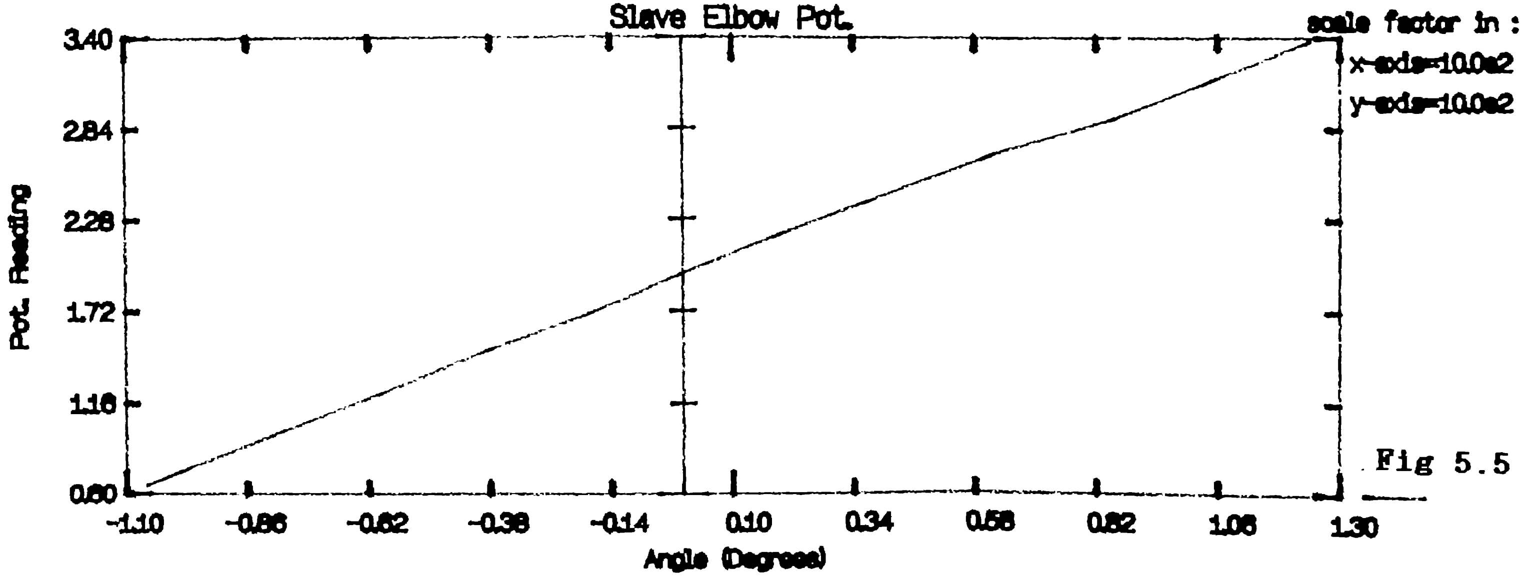


Fig 5.5

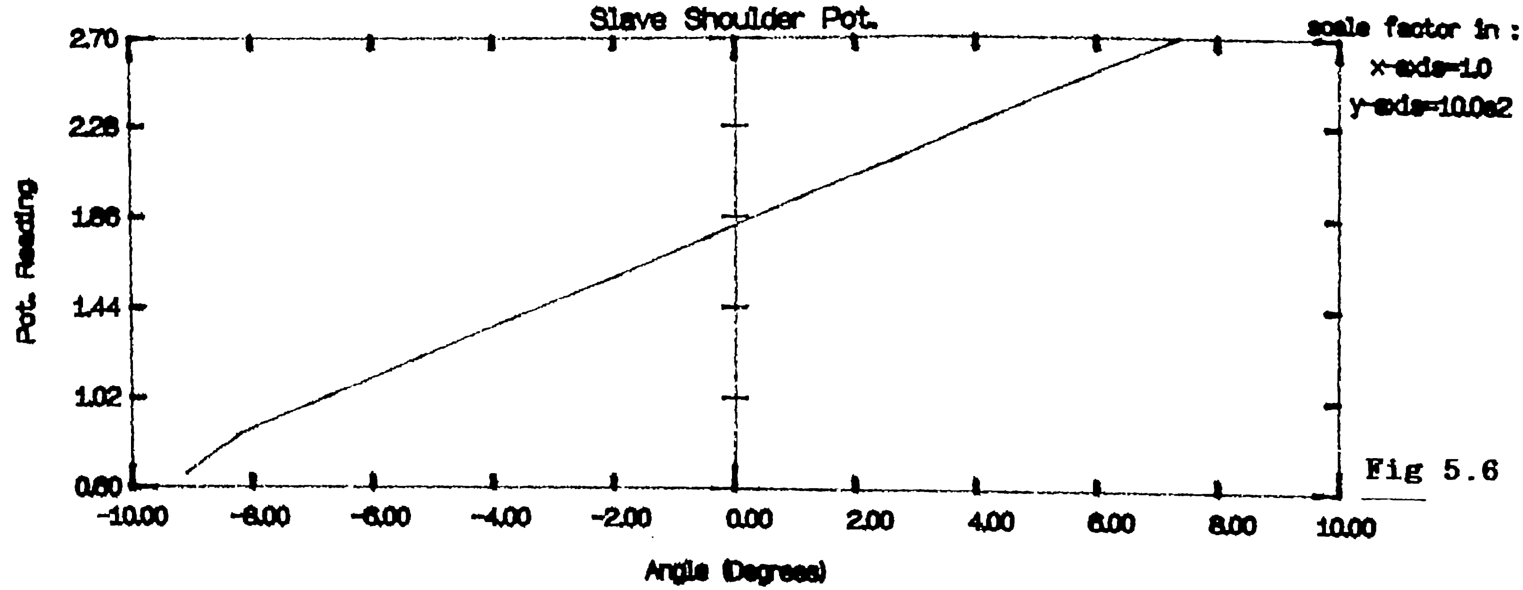


Fig 5.6

thus error $\theta_c^m - \theta_c^s$ can be derived from Eqn. 5 .2 as

$$\theta_c^m - \theta_c^s = \theta_c^m(1 - CAL\_\theta) + \theta_{min}^s - \theta_{min}^m .CAL\_\theta \qquad (5.3)$$

Thus it is evident that an approximately linear relation exists between error and the master joint variable.

5.3 DISCUSSION

The experimental results show that the potentiometers are indeed linear with respect to the measured joint angle values. The linearity is evident from the figures 5.1 to 5.6.

The error variation in the joint variable values is shown in figures 5.7 to 5.9. The error in joint variables of two is less when the joint motion is in the middle of the range. Error increases as the joints are moved in the range near the limits. Thus accuracy of the system is more if joint motions are limited to the middle range of joint limits.

Error in the waist joint varies from -11.0^o at one of the extremes to -1.5^o when the master joint angle reading is zero degrees, which lies exactly midway of the range. The error at shoulder joint shows similar variation. It varies from -26.5^o at one of the extremes to -1.5^o when the joint angle reading is 20^o, this once again is the middle portion of the joint variation. The elbow joint shows the best performance among the three. The error varies from 4^o at one of the limits to 0.0^o at zero master

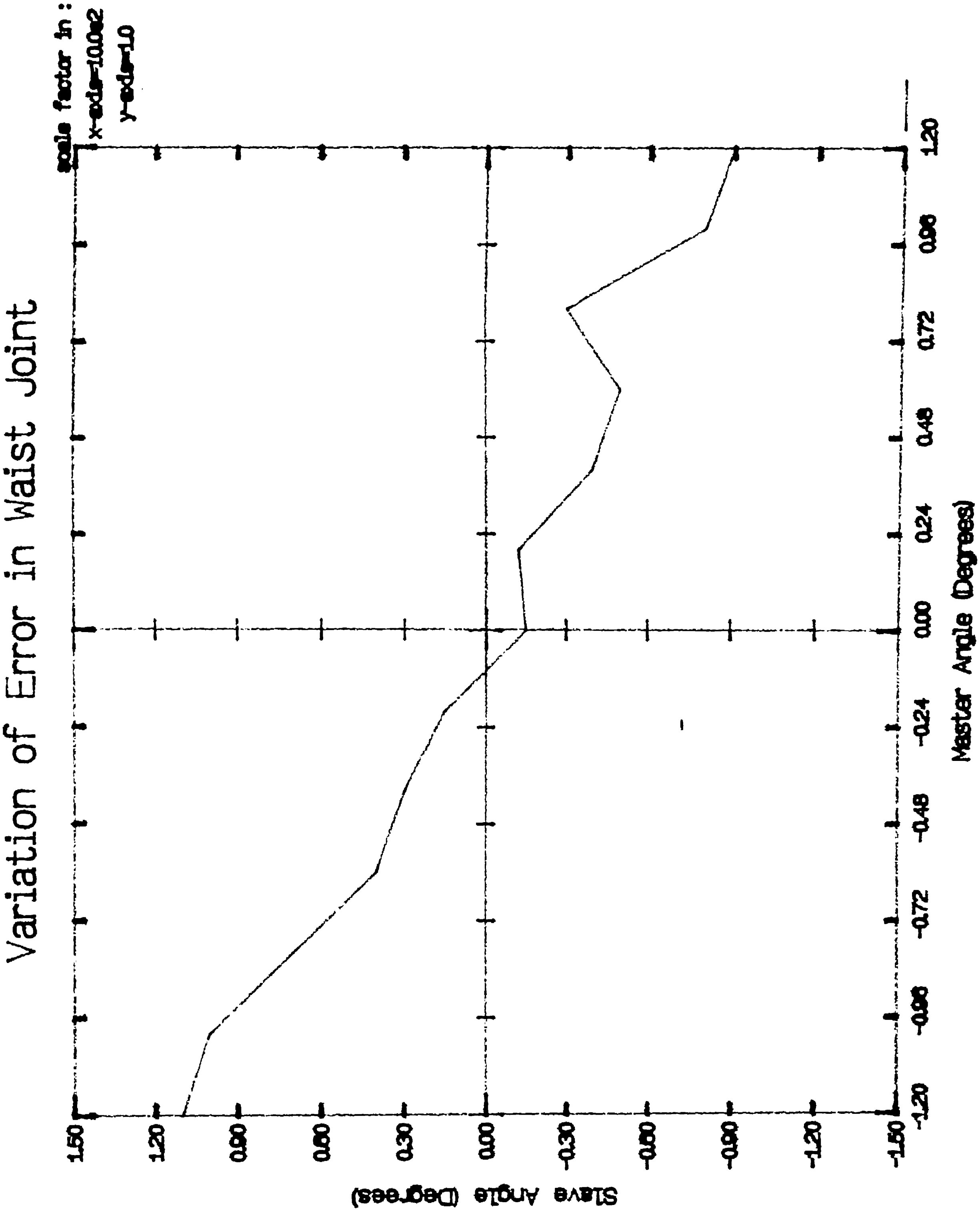

Variation of Error in Waist Joint
Master Angle (Degrees)
Slave Angle (Degrees)
scale factor in :
-1.20
-0.96
-0.72
-0.48
-0.24
0.00
0.24
0.48
0.72
0.96
1.20
1.50
0.90
0.60
0.30
-0.30
-0.60
-0.90
-1.50

Fig 5.7

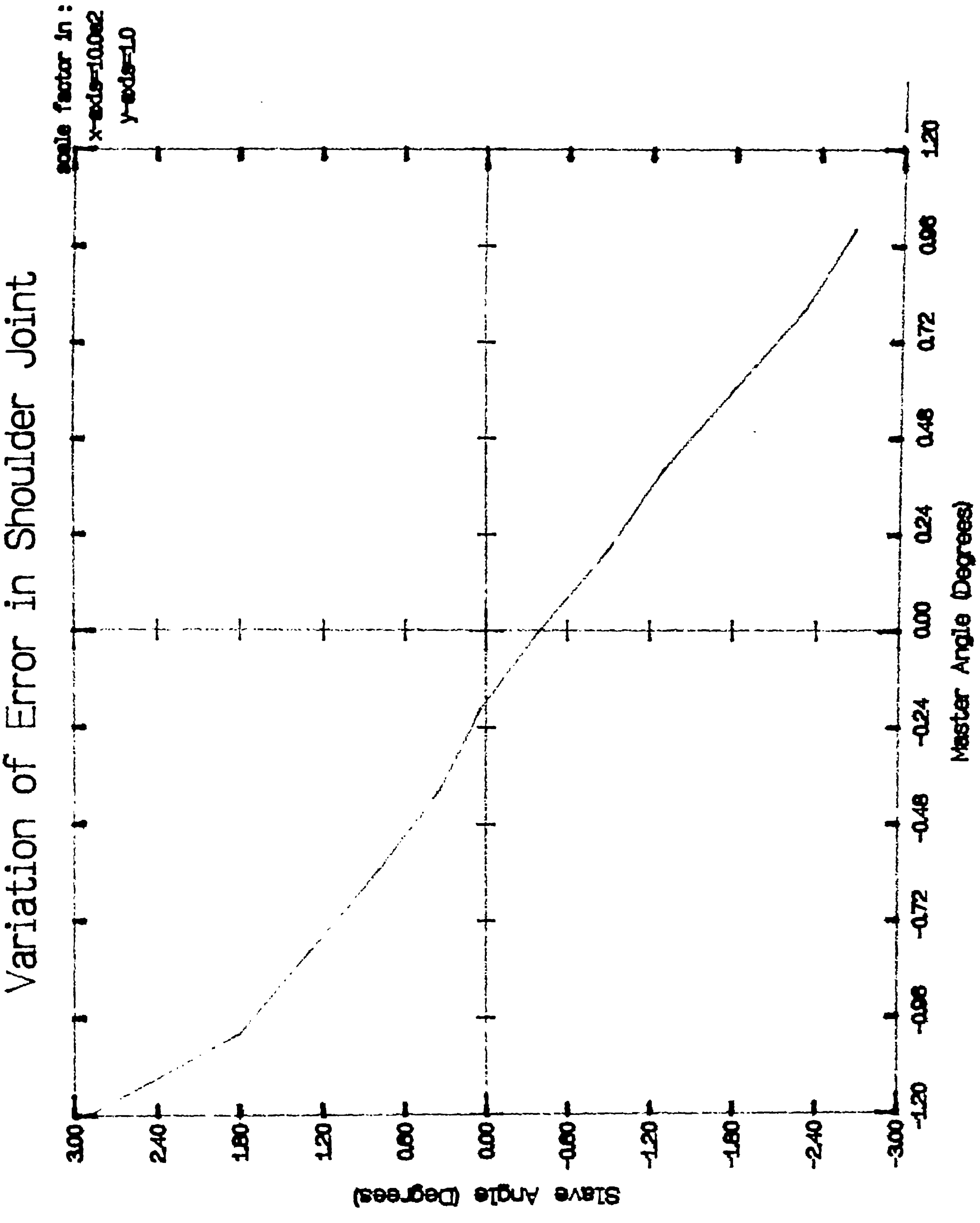
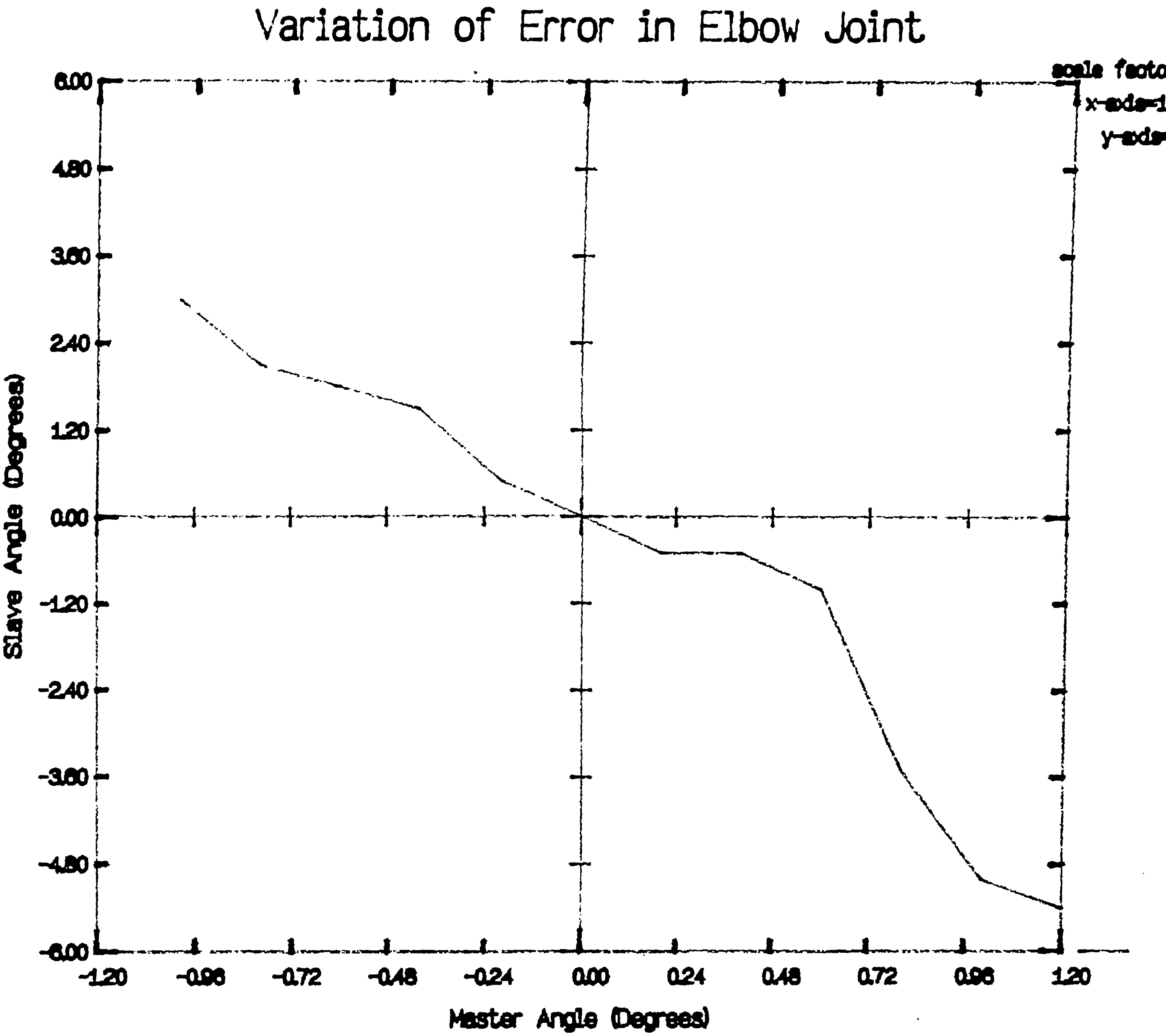

Variation of Error in Shoulder Joint
scale factor in :
x-axis=10.0e2
y-axis=1.0
-1.20
-0.96
-0.72
-0.48
-0.24
0.00
0.24
0.48
0.72
0.96
1.20
Master Angle (Degrees)
3.00
2.40
1.80
1.20
0.60
0.00
-0.60
-1.20
-1.80
-2.40
-3.00
Slave Angle (Degrees)

Fig 5.8



Variation of Error in Elbow Joint
6.00
4.80
3.60
2.40
1.20
0.00
-1.20
-2.40
-3.60
-4.80
-6.00
Slave Angle (Degrees)
-1.20
-0.96
-0.72
-0.48
-0.24
0.00
0.24
0.48
0.72
0.96
1.20
Master Angle (Degrees)

Fig 5.9

reading. The slave in this case closely follows the master.

5.4 CONCLUSIONS

With the limitations in fabrication and instrumentation, the master-salve robot performed well. With the implementation of the kinematic calibration procedure a very high accuracy at the end-effector is expected.

The algorithm for parameter estimation has been numerically verified. The simulation results show a good performance of the algorithm for a wide range of problems. Present work attempted to implement the parameter estimation algorithm to the teleoperated system. Due to limitations posed by lack of adequate instrumentation the objective could not be completely fulfilled. A complete step wise procedure has been laid out so that the calibration work can be completed once the hardware facilities are acquired.

The area of teleoperated system is gaining importance with the increasing remote applications. Importance of error analysis is well established for the industrial robots but with increasing need for remote manipulation it's application to systems like teleoperated robots is realized.

5.5 SUGGESTIONS FOR FURTHER WORK

The algorithm for parameter estimation currently

uses the Denavit-Hatenberg model. This model though suitable for obtaining forward and inverse kinematic solution is not suited for the purpose of error analysis. This has also been verified in current work that the Jacobian with respect to link offset parameter becomes singular for consecutive parallel axis. It is therefore suggested that certain modifications be done to avoid such a singularity. These modifications are suggested in [8].

The performance of master-slave can be improved by a complete error synthesis and calibration at kinematic parameter level. Optical encoders may be used instead of potentiometers. Once a better model and control strategy is adopted the need for visual feedback will be limited. This feedback may not be required if the task environment is completely known and it does not change significantly.

The slave arm can be fitted with the force sensors and master with the actuators so that a force feedback may also be available and hence complex operations can be performed. To reduce the delays in the operation of slave is also important to enhance the performance of the system.

REFERENCES

1. B. Mooring and S. Padavala, "The effect of kinematic model complexity on manipulator accuracy," in *Proc. IEEE conference Robotics Automation. (Pittsburg, PA)*, 1989.

2. C. H. Menq and J. H. Borm, "Statistical measures and characterization of robot errors." in *Proc. IEEE Int. Conf. Robotics Automation.*, p 926-930, 1988.

3. J. Chen and L. M. Chao, "Positioning error analysis for robot manipulators with all rotary joints, in *Proc. IEEE Int. Conf. Robotics Automation.*, p 1011-1016, 1986

4. J. M. Hollerbach, D. J. Bennet, "Autonomous calibration of single loop closed kinematic chains formed by manipulator with passive endpoint constraits," in *IEEE transactions on Robotics and Automation.*, vol.7, No.5, p 597-606 Oct 1991.

5. J. M. Renders, E. Rossingnol, "Kinematic calibration and geometrical parameter identification for robots," in *IEEE transactions on Robotics and Auto.*, vol.7, No.6, p 721-731, Dec 1991.

6. M. R. Driels and U. S. Pathre, " Vision Based Automatic Theodolite for Robot Calibration," in *IEEE transactions on RObotics and Automation*, vol.7, No.3, p 351-360 June 1991.

7. R. P. Judd and A. B. Knasinski, " A technique to calibrate industial robots with experimental verification," iin *Proc. IEEE Int. Conf. Robotics Automation.* (Raleigh, NC), p 175-182, 1987.

8. W. K. Veitscheggar and C. H. Wu, " Robot accuracy analysis based on kinematics," in *IEEE J. Robotics Automation*, vol.RA-2, p 171-180, Sept. 1986.

9. W. K. Veitschegger and C. H. Wu, " A method for calibrating and compensating robot kinematic errors, " in *Proc. IEEE Int. Conf. Robotics Automation.* (Raleigh, NC), p 39-44, 1987.

10. Z. S. Roth, B. W. Mooring, B. Ravani, "An Overview of Robot Calibration," in *IEEE J. of Robotics and Automation*, vol.RA-3, No.5, p 377-384, Oct. 1987.

11. C. H. An, C. G. Atkenson and J. M. Hollerbach, *Model-Based control of a Robot Manipulator*. Cambridge, MA : MIT Press, 1988.

12. J. Vertut and P. Coiffet, *Teleoperation and Robotics : Evalution and Development*, London, Kogenpage Ltd., 1985.

13. MA2000 *Software and Controller*, Tc Quipment International Ltd.

14. R. P. Paul, *Robot Manipulators : Mathematcs, Programming and Cotrol*, Cambridge, MA : MIT Press, 1982.

15. T. Yoshikawa, *Foundations of Robotics : Analysis and Control*, Cambridge MA : MIT Press, 1988.

16. Todd D. J., *Fundamentals of robot technology*, London, Kogenpage Ltd. 1986.

17. Unimate PUMA Robot *Volume 1 Technical Manual*, Unimation Inc., 1986.

APPENDIX I

SPECIFICATIONS OF MA 2000 ROBOT

A1.1 ROBOT MECHANISM

(i) Configuration : A revolute arm with 3 major and 3 minor axes.

(ii) Major axes : Waist, shoulder and elbow moving through 270° at maximum slew rate of 45° per second.

(iii) Wrist axes : Pitch, yaw, roll moving through 180° at a maximum slew rate of 90° per second.

(iv) Gripper : Pneumatically powered gripper attached to roll axis (a seperate power supply is required).

(v) Arm velocity : Nine programmable speeds, maximum speed 400 mm per seconds.

(vi) Load capacity : 1Kg dead lift at 480 mm from waist axis.

(vii) Drive system : Electric d.c. servo motors under closed-loop, 3 term control, with direct position feedback on each axis measured to 12 bit resolution.

(viii) Resolution : 1 part in 1000 over angular span on each axis.

(ix) Repeatability : $\pm$ 2 mm.

(x) Accuracy : $\pm$ 3 mm.

(xi) Joint position transducers : Plastic film potentiometers with linearity $\pm$ 0.25 %.

(xii) Sensor Supply : Arm pre-wired to accept microswitch or optical sensors at the gripper.

A1.2 CONTROLLER INTERFACE

(i) Interface: The robot is interfaced with the host computer through an Analog-to-Digital converter card.

(ii) I/O Ports: Two outputs each having relay contact switching having a capacity of 1 Amp. at 24 V DC. Four inputs each operating on connection to earth (ground) potential.

(iii) Safety: Incorporates an emergency stop button, "watch dog" timer and window detector circuit. Motor braking relay provides fall safe to set of major axis movement on interruption of power supply.

A1.3 MAIN OPERATING SOFTWARE

(i) Number of steps: The system can accomodate upto 100 taught steps in point-to-point operaton or one block of continuous path data can be memorised.

(ii) Step Commands: Position and speed mode of movements, I/O port state wait time and branch or jump instructions are possible.

(iii) Modes of Movement: The following are available at each step:

- No movement, control commands only.
- Move to absolute position.
- Move relative to last position.
- Search for input.
- Search and learn.
- Continuous path.
- Branch to user written "BASIC" subroutines.
- Report back on positional errors, actual position or motor powers.

(iv) Tutor Program: A 30 step interactive teach-and-learn sequence to familiarise user with the system.

A1.4 HOST COMPUTER

(i) Model: Acorn BBC Model B microcomputer complete with vide monitor (MA100).

APPENDIX II

EVALUATION OF JACOBIANS USING SCREW NOTATIONS

Evaluation of Jacobian can be done either by using differential homogeneous transformation or by using screw co-ordinate system. Derivation of the end-effector Jacobian using screw notations is as given below. For evaluating Jacobian with respect to theta, we have the diffrential relation

$$\delta\hat{x} = [\,J_\theta]\,\{\Delta\hat{\theta}\} \qquad (A2.1)$$

Or

$$\delta\hat{x} = \begin{bmatrix}\delta x\\ \delta y\\ \delta z\end{bmatrix} = \begin{bmatrix}J_{11} & J_{12} & \dots & J_{1n}\\ J_{21} & J_{22} & \dots & J_{2n}\\ J_{31} & J_{32} & \dots & J_{3n}\end{bmatrix}\begin{bmatrix}\Delta\theta_1\\ \Delta\theta_2\\ \vdots\\ \Delta\theta_n\end{bmatrix} \qquad (A2.2)$$

where n is the number of joints.

Considering a small variation $\Delta\theta_i$ in θ_i only, we have

$$\begin{bmatrix}\delta x\\ \delta y\\ \delta z\end{bmatrix} = \begin{bmatrix}J_{i1}\\ J_{i2}\\ \vdots\\ J_{in}\end{bmatrix}\cdot\{\Delta\theta_i\} \qquad (A2.3)$$

From figure A2.1 variation in θ_i is about $Z_{(i-1)}$ axis. Corresponding variation in end-effector position $\delta\hat{x}$ will be

$$\delta\hat{x} = [Z_{(i-1)} \times {}^{i-1}\hat{x}_E\,][\Delta\theta_i] \qquad (A2.4)$$

comparing equations (A2.3) and (A2.4) we have

$$_{i-1}[\,J^E_{i,\theta}] = [Z_{i-1} \times {}^{i-1}\hat{x}_E] \qquad (A2.5)$$

If the end-effector Jacobian is evaluated in the base frame then

$$ {}_{o}[J^{E}_{i,\theta}] = {}_{o}[A]^{i-1}[{}^{i-1}Z_{i-1} \times {}^{i-1}\hat{X}_{E}] \tag{A2.6} $$

Similarly expressions for Jacobians with respect to α, a and d can be written as follows

$$ {}_{o}[J^{E}_{i,\alpha}] = {}_{o}[A]^{i}[{}^{i}X_{i} \times {}^{i}\hat{X}_{E}] \tag{A2.7} $$

$$ {}_{o}[J^{E}_{i,d}] = {}_{o}[A]^{i-1}[Z_{i-1}] \tag{A2.8} $$

$$ {}_{o}[J^{E}_{i,a}] = {}_{o}[A]^{i}[X_{i}] \tag{A2.9} $$

For deriving Jacobian with respect to alpha and a it must be remembered that these parameters are located around X_i axis.

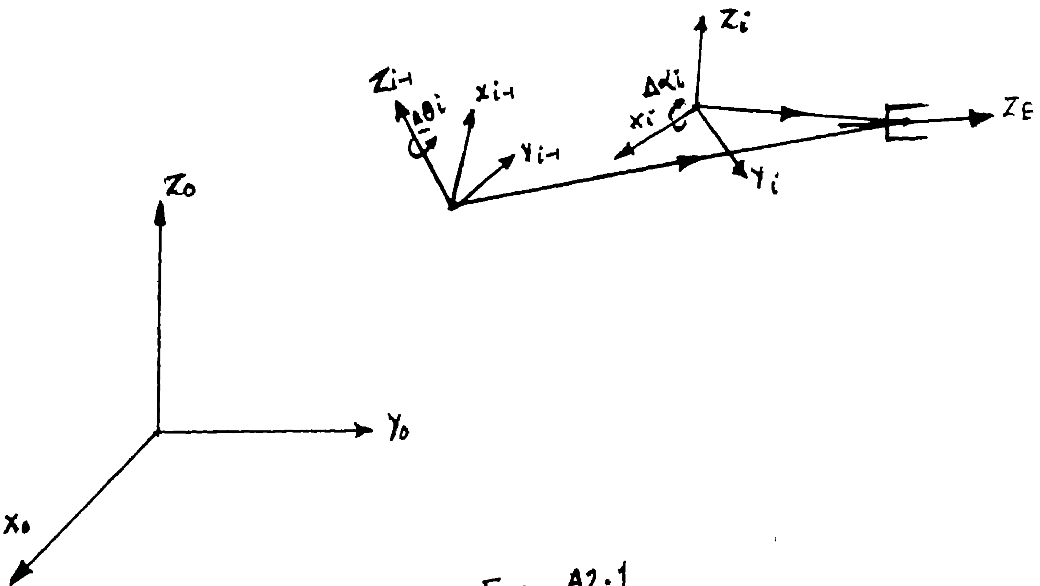


Fig. A2.1

APPENDIX III

PSEUDO-INVERSE

Inverse of a matrix plays an important role in linear algebra. They are particularly familiar as a means to express compactly the solution of simultaneous linear equations. However, a matrix has an inverse only when it is square and nonsingular. The pseudo-inverse is a generalization of the inverse to the case of singular and nonsquare matrices.

A n x m matrix A^+ is called pseudo-inverse of a m x n real matrix A if it satisfy the following four conditions

$$A.A^+.A = A, \qquad (A3.1)$$

$$A^+.A.A^+ = A, \qquad (A3.2)$$

$$(A.A^+)^T = A.A^+, \qquad (A3.3)$$

$$(A^+.A)^T = A^T.A. \qquad (A3.4)$$

Let a system of simultaneous linear equations be given by

$$A.x = b \qquad (A3.5)$$

where A is a known m x n matrix, b is a known m-dimensional vector,and x is an unknown n-dimensional vector. When A is square and nonsingular, the solution to above equation is known. If A is a nonsquare matrix then above equation is solvable but solution is not unique. Taking this into account if a more general problem of minimizing the Euclidean norm of error (Ax - b),

$$\|A.x - b\| = \sqrt{(A.x - b)^T(A.x - b)} \qquad (A3.6)$$

then the general solution to this problem is

$$x = A^+.b + (I - A^+ A).k \qquad (A3.7)$$

where k is an arbitrary n-dimensional vector.

This best approximate solution is also not unique, the one that

minimizes it's own norm is given by

$$x = A^{+}.b \tag{A3.8}$$

This can be proved as follows.We know that

$$A^{T}A\ A^{+} = A^{T} \tag{A3.9}$$

Using this relation, we can show that

$$b^{T}[I-(A\,A^{+})^{T}AA^{+}]b + [x-A^{+}b-(I-A^{+}A)\,k]^{T}A^{T}A[x-A^{+}b-(I-A^{+}A)k]$$

$$= x^{T}A^{T}Ax - 2x^{T}A^{T}b + b^{T}b$$

$$= \| Ax - b\|^{2} \tag{A3.10}$$

Since the first term on the left-hand side of equation A3.10 is independent of x, $\| Ax - b \|$ becomes minimum if and only if the second term is equal to zero, that is. if and only if

$$A[\,x - A^{+}b - (\,I - A^{+}A)k] = 0 \tag{A3.11}$$

or

$$x = A^{+}b + (\,I - A^{+}A)k \tag{A3.12}$$

Hence equation A3.7 is a solution to the problem for any k. If we assume k to be zero vector then

$$x = A^{+}b \tag{A3.13}$$